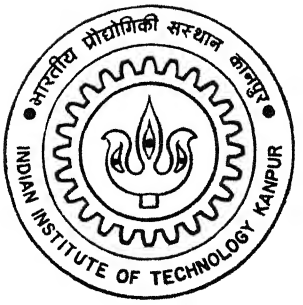
SET THEORETIC AND GRAPH BASED APPROACH FOR AUTOMATIC FEATURE RECOGNITION OF SHEET METAL COMPONENTS

by

RAGHAVENDRA JAGIRDAR

TH
ME/1995/D
J18S
ME
1995
D
JAG
SET



DEPARTMENT OF MECHANICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

FEBRUARY, 1995

SET THEORETIC AND GRAPH BASED APPROACH FOR AUTOMATIC FEATURE RECOGNITION OF SHEET METAL COMPONENTS

A Thesis Submitted

in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

by

Raghavendra Jagirdar

to the

DEPARTMENT OF MECHANICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

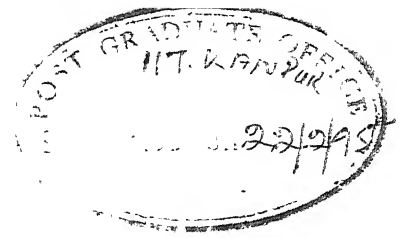
KANPUR 208016, INDIA

February, 1995

31 JUL 1996
CENTRAL LIBRARY
I. I. T., KANPUR
Acc. No. A. 121963

A121963

ME-1995-D-JAG-SET



CERTIFICATE

It is certified that work contained in the thesis entitled SET THEORETIC AND GRAPH BASED APPROACH FOR AUTOMATIC FEATURE RECOGNITION OF SHEET METAL COMPONENTS, by Raghavendra Jagirdar, has been carried out under our supervision and that this work has not been submitted elsewhere for a degree.

Prof. V. K. Jain

Department of Mechanical Engineering

I.I.T. Kanpur

Prof. J. L. Batra

Director

I.I.M. Lucknow

|| Poojyaya Raghavendraya Satyadharmaratayacha
Bhajatam Kalpavrukshaya Namatam Kamadhenave ||

DEDICATED TO

MY PARENTS

ACKNOWLEDGEMENTS

I feel great pleasure to express my sincere gratitude to my thesis supervisors Prof. V K Jain, and Prof. J L Batra for their expert guidance and wise counsel during the entire course. It had been indeed a great pleasure and privilege to work under their supervision.

I am grateful to Prof. S G Dhande for introducing me to this problem, for providing encouragement and for constructive criticism during the research.

I am obliged to Prof. G K Lal and Prof. Kripa Shanker for providing the computational facility in the CAM Lab, and for giving full flexibility to use it. I also, extend my gratitude to other faculty members of the Department of Mechanical Engineering– Manufacturing Science stream in particular, and Industrial and Management Engineering for their help in the present work.

My sincere thanks are due to Bajaj, Jha, Sandhya and other staff members of Manufacturing Science Lab for providing lively environment in the lab.

I shall remember my stay in the CAM Lab very much due to the research environment provided by Koshy and Reddy (Venkat), and for helping me bring the thesis to the present format. I am thankful to Satyadev for developing the flat pattern module and making it compatible with the prototype system developed during the research.

I shall always remember the benevolence and the homely atmosphere provided by the families of Prof. Jain, and Prof. Batra. I am also indebted to Deshpande (Vijay) and his family, and Gautam and his family for making my stay at the SBRA a very exiting and a memorable one.

I remember all the members of Kannada Sangha, especially Jayatheertha for making my stay at I I T – Kanpur a cheerful one.

I very much appreciate Mr. Vikram Kirloskar (Chairman and Managing Director) and the management of The Mysore Kirloskar Ltd., Harihar (India) for providing me leave to pursue higher studies, and for showing great interest in my research.

I am greatly indebted to Appa and Amma, who encouraged me and inspired me thorough out my life to do higher studies; to sister and her family, and brother for the constant encouragement they gave me.

I am beholden to my wife Savitha and little angel Rachita for providing cheerful and heartening environment to carry out research with zeal, and to forget the world on gloomy days during research.

R. Jagirdar

Contents

List of Tables xi

List of Figures xiii

Nomenclature xix

List of Publications xxiii

Synopsis xxv

1 Introduction and Literature Review 1

1.1 Introduction 1

1.2 Representation of components in a CAD database 4

1.2.1 Wireframe modeling 4

1.2.2 Surface modeling 6

1.2.3 Solid modeling 6

1.2.4 Feature based modeling 13

1.3 Literature review 15

1.3.1 Feature recognition in general 15

1.3.2 Feature recognition for sheet metal components 24

1.4 Scope and statement of purpose 27

1.5 Organization of the dissertation 30

2 Classification Systems for Pressworking Features 31

2.1 Introduction 31

2.2 General Classification Systems for Sheet Metal Components and Processes 33

2.2.1 Classification systems for sheet metal components 33

2.2.2 Classification system for pressworking processes 33

2.2.3 Proposed classification systems for pressworking processes and features 35

2.3 Definitions 38

2.4 Classification system for shearing operations and features 41
2.5 Classification system for forming operations and features 44

3 Graph Generation and Subgraph Identification **51**
3.1 Introduction 51
3.2 Definitions 53
3.3 Types of entities used and assumptions made 55
3.4 Identification of duplicate vertices 57
3.5 Labelling of nodes and edges in the lexicographical order 60
3.5.1 Lexicographical ordering of vertices 61
3.5.2 Lexicographical ordering of edges 61
3.6 Connectedness and subgraph identification 65
3.6.1 Fusion methodology 65
3.7 Identification of nature of input to the system 70

4 Generation and Identification of Feature Sets **77**
4.1 Introduction 77
4.2 Classification of the feature set 77
4.3 Minimum enclosing rectangle (MER) for features 81
4.4 Ray containment test for features 82
4.5 Identification of various feature sets 86
4.5.1 Identification of raw material feature set 88
4.5.2 Identification of boundary and inside feature sets 89
4.5.3 Implementation 90
4.6 Identification of component set 90

5 Shearing Feature Recognition System **93**
5.1 Introduction 93
5.2 Generation of adjacency component directional relationship (ACDR) table 94
5.2.1 Determination of direction for a component with respect to other components 94
5.2.2 Identification of adjacent components 97
5.3 Identification of raw material type 102
5.4 Identification of layout type and the direction of layout 106
5.5 Recognition of shearing features 112
5.5.1 Recognition of inside shearing features 112
5.5.2 Recognition of boundary shearing features 113

5.5.3 Recognition of outside features 128

6 Forming Feature Recognition System **139**

6.1 Introduction 139

6.2 Identification of parent and child subgraphs 140

6.2.1 Identification of parent subgraph 143

6.2.2 Identification of child subgraphs 144

6.3 Identification of loops in a component graph 144

6.3.1 Determination of number of independent loops in a component graph 150

6.3.2 Calculation and identification of loops in a parent subgraph 153

6.3.3 Identification of loops in child subgraphs 160

6.3.4 Identification of child subgraph loop present in the parent subgraph loop 161

6.4 Recognition of a curved plane in the component 165

6.5 Identification of adjacent planes 165

6.6 Cross-bend feature extraction 170

6.7 Recognition of forming features 170

6.7.1 Characteristics of forming features 173

6.7.2 Feature recognition 180

6.8 Flat pattern development for a component 181

6.8.1 Calculation of bend allowance in a component and implementation 184

6.8.2 Flat pattern development and implementation 188

6.9 Recognition of shearing features 193

6.9.1 Creation of the 2–D component from the flat pattern of a 3–D component 194

6.9.2 Recognition of inside features of a 3–D component from its flat pattern 196

6.9.3 Recognition of boundary and outside features of a 3–D component from its flat pattern 196

7 Prototype System and Implementation **197**

7.1 Introduction 197

7.2 Implementation 202

8 Conclusions and Scope for Future Research **243**

8.1 Conclusions 243

8.2 Scope for future research 245

A Definitions of Shearing Operations **247**

B Definitions of Forming Operations 251

C Definitions of Set theory 257

D Ray containment test 261

References 263

Bibliography 271

List of Tables

2.1 Design and manufacturing attributes typically included in classification systems (Groover, 1992). 32
2.2 Classification system of sheet metal pressworking processes (Lascoe, 1988). . . 34
2.3 Sheet metal classification system based on the contour of finished parts (Sachs, 1966). 45
2.4 Classification system for sheet metal forming processes, based on the contour of the finished parts (Altan et al., 1983). 46

3.1 First and second vertices of the edges are labeled according to the lexicographically ordered vertex set. 64
3.2 First and second vertices of the edges are rearranged. 64
3.3 Vertices and edges present in the subgraphs of the 3–D component (Fig. 3.10(a)). 70

4.1 List of vertices after lexicographical ordering for component layout shown in Fig. 4.1. 79
4.2 Vertices present in various feature sets. 80
4.3 Coordinates of vertices belonging to MER of features. 82
4.4 Various partition sets of the feature set, and inside features belonging to a boundary feature. 91
4.5 Component set, and features present in a component. 92

5.1 Rules for identifying the direction of an adjacent component c_k' with respect to the base component c_k. 95
5.2 List of vertices after lexicographical ordering. 97
5.3 Vertices and their degree in the component graph. 98
5.4 Features present in various feature sets. 98
5.5 List of pseudo vertices present in boundary features. 98
5.6 Rules for determining the intersection between MER of components. 99
5.7 Rules for finding vertices of MRBC between components. 100
5.7 Rules for finding vertices of MRBC between components (continued). 101

5.8 Rules for determining whether the base vertex falls within the segment of the surface of the adjacent component under consideration or not. 117
5.9 Cutoff features present in the component layout given in Fig. 5.8. 118
5.10 Notching features present in the component layout given in Fig. 5.2. 125
5.11 Parting features present in the component layout given in Fig. 5.2. 129
5.12 Slitting features present in the component layout given in Fig. 5.7. 131
5.13 Slitting features present in the component layout given in Fig. 5.4. 135
5.14 Shearing features present in the component layout given in Fig. 5.7. 137

6.1 Parent and child subgraphs of a component graph having the maximum projected length in all three directions. 150
6.2 For Fig. (6.2) : (a) Vertices and edges of subgraphs present in the component graph, and (b) Parent and child subgraphs of a component graph having maximum projected length in X and Y directions. 151
6.3 For the 3–D component shown in Fig. 6.3, vertices and edges of subgraphs present in the various subgraphs, and parent and child subgraphs, present in the component are given. 152
6.4 Edges present in different loops of the parent subgraph, and common edges present in various loops. 160
6.5 Curved planes present in the component graph. 165
6.6 Bend allowance determined for a curved plane of the component graph shown in Fig. 6.3(a). 189

7.1 DXF output file format representing the various sections present in the file. . . 198
7.2 Explanation of an entity format given in DXF output file. 199
7.3 List of vertices after lexicographical ordering. 210
7.4 List of edges after lexicographical ordering. 211
7.5 List of vertices after lexicographical ordering. 218
7.6 List of edges after lexicographical ordering. 219
7.7 List of vertices after lexicographical ordering. 225
7.8 List of edges after lexicographical ordering. 226
7.9 List of vertices after lexicographical ordering. 233
7.9 List of vertices after lexicographical ordering (continued). 234
7.10 List of edges after lexicographical ordering. 235
7.10 List of edges after lexicographical ordering (continued). 236

List of Figures

1.1 Society of Manufacturing Engineers CIM wheel (Rehg, 1994). 2
1.2 The enterprise areas (Rehg, 1994). 3
1.3 Wireframe models : (a) prismatic, (b) 2–D sheet metal part nested layout, and (c) 3–D sheet metal component. 5
1.4 Data structure of a wireframe model : (a) Stepped cube, and (b) Data structure of the stepped cube giving edge information. 7
1.5 Wireframe model for a stepped cube. 8
1.6 Ambiguous wireframe models : (a) Tesserac, (b) (c) and (d) valid planar-faced solids (Agarwal and Waggenspack Jr., 1992). 8
1.7 Boundary faces of a cube(Agarwal and Waggenspack Jr., 1992). 8
1.8 Data structure of a surface model : (a) Stepped cube, and (b) Data structure of the stepped cube. 9
1.9 Types of polyhedral objects (Zeid, 1991). 10
1.10 General data structure for a boundary modeling (Zeid, 1991). 11
1.11 A typical solid and its building primitives : (a) Typical solid, and (b) Primitives (Zeid, 1991). 11
1.12 CSG graph of a typical solid (Zeid, 1991). 12
1.13 Data structure of a typical solid Primitives (Zeid, 1991). 12
1.14 Schematics of feature-definition approaches : (a) Interactive features definition, (b) Automatic feature definition, and (c) Design by features (Shah, 1991). . . 13
1.15 4-level data structure of a feature model (Shah and Rogers, 1988). 14
1.16 Various types of features : (a) Functional features, (b) Form features, and (c) Manufacturing features. 16
1.17 Pattern primitives. 17
1.17 A simple hole drawing. 17
1.19 An architecture for a syntactic pattern recognition based feature recognizer. . . 18
1.20 State transition diagram for step and slot (Chang, 1990). 19
1.21 Derivation of ASV series. 20
1.22 Example of AAG for a part (Joshi and chang, 1988). 22

1.23 AAG of feature instances (Joshi et al., 1988). 22
1.24 Wireframe model representation of sheet metal component and part layout in the present work: (a) 2–D part layout, and (b) 3–D component. 28

2.1 Feature classification for sheet metal application (Nnaji et al., 1991). 36
2.2 General classification system for pressworking processes. 37
2.3 General classification system for pressworking features. 38
2.4 Representation of sheet metal components in a wireframe model : (a) Plan of a 2–D nested layout, (b) A 2–D component, and (c) Mean plane of a 3–D component. 39
2.5 Classification system for shearing operations. 42
2.6 Classification system for shearing features. 43
2.7 Classification system for forming operations. 47
2.8 Classification system for forming features. 49

3.1 Nested layout of a 2–D component : (a) Plan view, and (b) Layout graph. . . 52
3.2 A 3–D sheet metal component : (a) Mean plane of the component, and (b) Component graph. 54
3.3 Entities names, and their geometrical and graphical representation. 56
3.4 Arc measurement direction. 58
3.5 Component graph representing vertices and edges : (a) Graph before duplicate vertices are identified, and (b) Graph after duplicate vertices are identified, deleted and labeled. 59
3.6 (a) Randomly ordered vertex set and its coordinate values, (b) Lexicographically ordered vertex set, and (c) Component graph and vertex set after labeling of lexicographically ordered vertex set again. 62
3.7 (a) Lexicographically ordered edge set, and (b) Component graph and edge set after relabeling the above edge set. 63
3.8 Fusion of vertex v_3 and v_4. 66
3.9 Flowchart for identifying connectedness and subgraphs in a component/part layout graph. 67
3.10 (a) Graph of the 3–D component with vertices and edges labeled according to the lexicographical order, and (b) Vertex adjacency matrix for the component graph. 69
3.11 (a) A 2–D component nested layout graph, and (b) A 3–D component graph. . 72
3.12 Flowchart for identifying the nature of the drawing given as input to the system. 74

3.12 Flowchart for identifying the nature of the drawing given as input to the system (continued). 75

4.1 A 2–D dissimilar component nested layout : (a) Nested layout, and (b) Nested layout graph. 78
4.2 Ray containment test to determine whether a point (P) is contained in a polygon or not. 83
4.3 Various ways a horizontal ray intersects with edges of a polygon : (a) Intersecting an edge, (b) Passing along an edge, and (c) Passing through a vertex. 84
4.4 Ray containment test to determine whether a feature is contained in another feature or not. 86

5.1 Adjacency directions in accordance to cartesian octants. 94
5.2 Component nested layout graph : (a) Layout of a 2–D component, and (b) Component layout graph representing pseudo vertices and features. 96
5.3 Flowchart for determining adjacency component relationship matrix and ACDR table. 103
5.3 Flowchart for determining adjacency component relationship matrix and ACDR table (continued). 104
5.4 Dissimilar components nested layout : (a) Nested layout, (b) Layout graph, and (c) component set and features present in each component. 105
5.4 Dissimilar components nested layout : (a) Nested layout, (b) Layout graph, and (c) component set and features present in each component (continued). 106
5.5 For figure 5.4 : (a) Adjacency component relationship matrix, (b) ACDR table, and (c) Subgraphs of components present in the directions of 1 and 3, 2 and 4, and 1, 3, 5 and 7. 107
5.6 Dissimilar components nested layout having intersecting MER of components : (a) Nested layout, (b) Layout graph and MER of components, and (c) ACDR table. 108
5.7 A similar type of components nested layout : (a) Nested layout, (b) ACDR table, and (c) Subgraphs of components present in the directions of 1 and 3, 2 and 4, and 1, 3, 5 and 7. 110
5.8 A similar type of components nested layout having cutoff feature : (a) Nested layout, (b) ACDR table, and (c) Subgraphs of components present in the directions of 1 and 3, 2 and 4, and 1, 3, 5 and 7. 114
5.9 Modes of a vertex of the base component falling on the adjacent component surface (for similar type of components). 116

5.10 Flowchart for recognizing cutoff feature. 119
5.10 Flowchart for recognizing cutoff feature (continued). 120
5.11 (a) Minimum enclosing rectangle for a component, and (b) Representation of vertices and pseudo vertices, along with their convexity and concavity. 122
5.12 Various tools that can be used for blanking a component :(a) Representing MER of a component and notching features, (b) Blanking tool having same shape and size as that of the MER, and (c) Another shape of the blanking tool. 124
5.13 Flowchart for recognizing notching features. 126
5.14 Parting features for a layout in the direction of X axis. 127
5.15 Flowchart for recognizing slitting features. 132
5.15 Flowchart for recognizing slitting features (continued). 133
5.15 Flowchart for recognizing slitting features (continued). 134

6.1 Flowchart for identifying parent and child subgraphs. 145
6.1 Flowchart for identifying parent and child subgraphs (continued). 146
6.1 Flowchart for identifying parent and child subgraphs (continued). 147
6.2 A 3–D component having maximum projected length in two directions : (a) Component, and (b) Component graph with lexicographically ordered vertices and edges . . 148
6.3 A 3–D component having 3–D and 2–D child subgraph features : (a) Component, and (b) Component graph with lexicographically ordered vertices and edges 149
6.4 For the 3–D component given in Fig. 6.3(a) : (a) Parent subgraph of the component, and (b) Edge adjacency (EA) matrix of parent subgraph along with degree of an edge and degree of a loop. 155
6.5 A 3–D component having edges in a same plane, but forming a loop and a path: (a) Graph showing edges e_1, e_2, e_3, e_4 and e_7 located in a plane, and (b) Connectedness of edges in the plane under consideration. 157
6.6 Steps in determining independent loops present in the parent subgraph. 158
6.7 (a) Child subgraph g_2 of the component graph and EA matrix along with degree of edge and loop, and (b) Edges present in different loops, and child subgraph loop located in the parent subgraph. 162
6.8 (a) Child subgraph g_3 of the component graph and EA matrix along with degree of edge and loop, and (b) Edges present in different loops, and child subgraph loop located in the parent subgraph. 163
6.9 Flowchart for identifying child subgraph loop located in the parent subgraph loop. 164
6.10 Flowchart for identifying curved plane in a component graph. 166

6.11 Flowchart for determining adjacency plane relationship (APR) matrix for a component graph. 167
6.11 Flowchart for determining adjacency plane relationship (APR) matrix for a component graph (continued). 168
6.12 (a) APR matrix of the parent subgraph, and (b) APR matrix of the child subgraph. 169
6.13 (a) A component having cross-bend features, and (b) Component graph. . . . 171
6.14 (a) The component graph after removing common edges between planes having cross-bend features, and (b) Vertices and edges present in cross-bend features. 172
6.15 Graphical representation, and geometrical and topological information of forming features present in a single plane of the parent subgraph. 175
6.16 Graphical representation, and geometrical and topological information of forming features present in a few multi-planes of the parent subgraph. 176
6.17 Graphical representation, and geometrical and topological information of forming features present in all planes of the parent subgraph, and in a few planes of parent and child subgraphs. 177
6.18 (a) Bending terminologies, and (b) Bending stresses. 183
6.19 Location of important fibers in bending (Prasad and Somasundaram, 1993). . 185
6.20 Flowchart for calculating bend allowance in the curved planes of a component. 186
6.20 Flowchart for calculating bend allowance in the curved planes of a component (continued). 187
6.21 (a) Mean axis of a bend, and (b) Flattening process along the mean axis of the bend. 191
6.22 Flat pattern development of the 3–D component shown in Fig. 6.3. 193
6.23 Creation of boundary and inside features from the flat pattern development given in Fig. 6.22. 195

7.1 Various Functions of the prototype system. 199
7.2 Flowchart representing the flow of control for the prototype system. 200
7.2 Flowchart representing the flow of control for the prototype system (continued). 201
7.3 Various functions of shearing feature recognition system. 203
7.4 Operation based data structure for recognizing shearing features. 204
7.5 Various functions of forming feature recognition system. 205
7.6 Operation based data structure for recognizing forming features. 206
7.7 Bezel layout of a watch manufactured at HMT (India) : (a) Drawing of the bezel nested layout given as input to the system, and (b) Bezel layout giving a part of information about the output for easy reference. 209

7.8 Dissimilar components used in the manufacturing of VMC–345 : (a) Drawing of the layout given as input to the system, and (b) Dissimilar component layout giving a part of information about the output for easy reference. 217
7.9 A 3–D component (Example–3) : (a) Drawing of the component given as input to the system, and (b) Subgraph and plane numbers are represented for easy reference. 223
7.10 Flat pattern development of the 3–D component shown in Fig. 7.9. 224
7.11 A complex 3–D component : (a) Drawing of a modified part of sun microphone box given as input to the system, and (b) Representation of graph and plane numbers for easy reference. 231
7.12 Flat pattern development of the 3–D component shown in Fig. 7.11. 232

C.1 Basic set operations 258

Nomenclature

a	boolean variable.
b	element of the boundary feature set.
bl	bulge of an arc.
bp	bend plane in a component graph.
B	boundary feature set.
BA	bend allowance.
c	element of the component set.
c_x, c_y, c_z	directional cosines.
C	component set.
CC	components absent in the columns of a layout.
CV	coefficient in a planar equation.
$C_{1,3}$	components present in the direction of 1 and 3.
$C_{1,3,5,7}$	components present in the direction of 1, 3, 5, and 7.
$C_{2,4}$	components present in the direction of 2 and 4.
$C_{2,4,6,8}$	components present in the direction of 2, 4, 6, and 8.
d	element of the directional set.
$d(e)$	degree of an edge.
$d(v)$	degree of a vertex.
D	directional set.
e	element of the edge set.
e_{in}	surface strain at innermost fiber.
e_{out}	surface strain at outermost fiber.
E	edge set, whose elements are lexicographically ordered.
f	element of the feature set.
fp	flat plane in a component.
F	feature set.
g	element (subgraph) of a component graph.
G	a 3–D component graph.
G_c	child subgraphs present in a component graph.
G_p	parent subgraph present in a component graph.
G_X	subgraphs having maximum projected length along X axis.
G_Y	subgraphs having maximum projected length along Y axis.
G_Z	subgraphs having maximum projected length along Z axis.
G_{XY}	subgraphs having maximum projected length along X and Y axes simultaneously.

G_{YZ}	subgraphs having maximum projected length along Y and Z axes simultaneously.
G_{ZX}	subgraphs having maximum projected length along Z and X axes simultaneously.
G_{XYZ}	subgraphs having maximum projected length along X, Y and Z axes simultaneously.
$G_{1,3}$	subgraphs of components present in the direction of 1 and 3.
$G_{1,3,5,7}$	subgraphs of components present in the direction of 1, 3, 5 and 7.
$G_{2,4}$	subgraphs of components present in the direction of 2 and 4.
$G_{2,4,6,8}$	subgraphs of components present in the direction of 2, 4, 6 and 8.
h	head vertex of a component.
I	set of subscripts of the lexicographically ordered and re-labeled vertex set.
J	set of subscripts of the feature set or component graph.
K	set of subscripts of the component set.
l	loop in a component graph.
$l(e)$	loop degree of an edge.
$lc(e)$	loop count of an edge.
len	projected length of feature or subgraph along an axis.
L	loops present in a component graph.
M	set of subscripts of the edge set.
$[M]_{p_0}$	complete transformation matrix at a point.
$[M]_{BA}$	bend allowance matrix.
n	element of the inside feature feature set.
nc	number of components present in a 2–D nested layout.
nf	notching features.
N	inside feature set.
Nc	total sum of values assigned to the total number of intersections occurred between horizontal ray and line segment of a boundary feature.
NV	coefficient of a plane equation.
p	pseudo vertex.
r	element of the raw material feature set.
rad	radius of an arc or a circle.
R	raw material feature set.
R'	complementary set of the raw material feature set.
RC	components absent in the rows of a layout.
R_{in}	inner radius of bend.

R_m	radius of mean fiber.
R_{neu}	radius of neutral fiber.
R_{out}	outer radius of bend.
R_{un}	radius of unstretched fiber.
R_{unnew}	radius of new unstretched fiber.
$[R_X]$	rotation matrix about X axis.
$[R_Y]$	rotation matrix about Y axis.
$[R_X]^{-1}$	inverse rotation matrix about X axis.
$[R_Y]^{-1}$	inverse rotation matrix about Y axis.
$[R_\theta]$	rotation matrix about an arbitrary axis.
shr	shearing range.
slr	slitting range.
S	number of subgraphs present in a component graph.
Sl	slug length.
Sw	slug width.
t_d	deformed sheet thickness.
t_{dnew}	new deformed sheet thickness.
t_{in}	initial sheet metal thickness.
$[T]$	translation matrix.
v	element of a vertex set.
V	vertex set, whose elements are lexicographically ordered.
x, y, z	coordinates of a vertex.
X, Y, Z	represent X, Y, and Z axes respectively.
$Z1X$	feature set whose elements have maximum projected length along X axis.
$Z1Y$	feature set whose elements have maximum projected length along Y axis.
$\emptyset$	NULL set.
μ	scalar constant.
ν	scalar constant.
$\psi(v)$	isomorphsim of a vertex.
σ_y	yield stress.
θ	included angle of an arc.

Subscripts

c	for center of an arc or a circle.
co	for coil type of raw material.
col	represents the col^{th} column of a subgraph.

d	for sets determined from the CAD database without further processed.
d'	for sets obtained after identifying repeated vertices.
e	end of an entity.
i	represents the i^{th} element of the vertex set V.
i'	represents the i'^{th} element of the vertex set V, such that $i' \neq i$.
j	represents the j^{th} element of the feature set F.
j'	represents the j'^{th} element of the feature set F, such that $j' \neq j$.
k	represents the k^{th} element of the component set C.
k'	represents the k'^{th} element of the component set C, such that $k' \neq k$.
lt	left side of the column.
lo	for lexicographically ordered set, before labeling again.
lw	lower side of the row.
m	represents the m^{th} element of the edge set E.
m	represents the m'^{th} element of the edge set E such that $m' \neq m$.
max	maximum value.
min	minimum value.
rt	right side of the column.
row	represents the row^{th} row of a subgraph.
s	start of an entity.
st	strip type of raw material.
up	upper side of the row.

Superscripts

x	represents x coordinate value.
y	represents y coordinate value.
z	represents z coordinate value.

Vectors

$\vec{D}$	directional vector.
$\vec{HL}$	horizontal ray emerging from a point.
$\vec{i}, \vec{j}, \vec{k}$	unit vectors.
$\vec{L}$	line vector.
$\vec{NV}$	normal vector.
$\vec{NV}_{ref}$	normal vector at a reference point.
$\vec{V}_{arb}$	arbitrary vector.
$\vec{V}_{dev}$	development vector.

List of Publications

1. Characterization of shearing features for sheet metal components in 2–D layout, *International Journal of Production Research*, accepted for publication.

2. Feature recognition methodology of shearing operations for sheet metal components, *Computer Integrated Manufacturing Systems*, in print.

3. Characterization of forming features for sheet metal components (manuscript under preparation).

4. Feature recognition methodology of forming operations for sheet metal components (manuscript under preparation).

5. Classification system of pressworking operations and features (manuscript under preparation).

SYNOPSIS

Introduction

Sheet metal components form a significant part of manufacturing activity, and they are widely used in different types of industries like aerospace, machine tools, electronics, refrigeration and air conditioning, etc. Manufacturing processes of these components vary in difficulty from simple to complex operations. At present, in most of the industries, type of pressworking operations are determined by manually transferring the design information into manufacturing information. Also, process plans for sheet metal components, codification of components (from manufacturing point of view), generation of G and M codes for CNC machine tools to manufacture the components, mapping form features for the development and inspection of tool shapes, etc., are all done manually. To overcome the difficulties and limitations inherently associated with human beings, research work is going on to obtain manufacturing information directly and automatically from the design information through the method called *feature recognition.*

Feature recognition of a part consists of two major tasks; extracting features from the component, and recognizing them. Review of the current literature shows that the *approaches* used for feature recognition in sheet metal components are:

1. Logic approach
2. Graph based approach
3. Feature based approach

Logic approach is associated with an expert system. The topological structure of the manufacturing features is written in terms of logic rules, and the programming language usually used is either Prolog or Lisp. Logic approach is usually used for 2–D and 2-1/2 D parts, as it is easy to write rules for these type of parts. In *graph based approach*, firstly an attributed adjacency graph (AAG) is developed for the part using boundary representation. In AAG, nodes represent faces of a component, and edges represent the connectivity between them. An attribute of '1' or '0' is assigned to indicate whether faces sharing the common edge form a convex or concave edge, respectively. By applying heuristics, unimportant nodes are deleted from the graph, and the remaining graph is used

for identification of features in a hierarchical order. Geometry of a feature is not considered for identifying features in this approach. Therefore, features based on the geometry require one more step (i.e. geometric reasoning) to recognize them. In *feature based approach*, a component is created in CAD system by features. As such, the information on features becomes automatically available from the CAD database. Features, in the feature based design are associated with geometric attributes and some specific feature attributes, which are process specific.

Review of the literature shows that by combination of geometric information about the component and graph based approach provides a good platform to recognize features of the components created by solid modeler. Further, the current literature indicates that most of the researchers have used solid modeling for feature recognition of sheet metal parts and have converted the component created by solid modeler into the foil type of component by neglecting its thickness, and pressworking features are identified from these foil type of components. However, in practice, most of the industries use wireframe modeling to represent the sheet metal components because wireframe modeling is easy to create and it can represent the sheet metal components unambiguously.

The literature review also reveals that in the feature recognition process, in general, form features are recognized first from the design database, and then manufacturing features are recognized. In case of sheet metal components, form features produced during pressworking operations depend on tool shape, size and force applied on it, and not on the relative movement between the tool and workpiece as in case of conventional metal cutting operations. Therefore, in the present work, manufacturing features are identified directly from the design database. Further, pressworking operations like cutoff, notching, parting, shearing and slitting are also dependent on the parameters like, type of raw material (coil, sheet and strip), type of layout and direction of layout. These operations and parameters are not considered in most of the research work presented in the literature on recognition of pressworking features for sheet metal components.

Statement of purpose

The *overall objective* of the present work is to recognize the pressworking (manufacturing) features directly and automatically from the CAD database generated for a 3–D component and/or a 2–D component layout created by wireframe modeler. The process of recognizing features involves the development of a set of generalized principles by reasoning and relating them geometrically and topologically using set theory and graph theory concepts.

The *specific objectives* of the present work can be stated as follows:

1. To recognize pressworking features, by the proposed methodology, it is required to determine the *classification system of features* for developing a set of principles for characterizing them. Characterization of features helps in identifying them automatically from the CAD database in the hierarchical way. To classify pressworking features, the pressworking operations have to be classified first, as the features are directly related to the operations. Therefore, a classification system for operations is proposed initially which is subsequently extended for classification of features.

2. To identify shearing and forming features, along with the geometrical parameters, directly from the CAD database of the component and/or part layout created flexibly in wireframe modeling. The features so identified will give the alternative methods by means of which a component can be manufactured from the raw material.

3. To develop the flat pattern for a formed component by taking thinning effect into consideration for recognizing shearing and forming features.

4. To show the capabilities of the proposed methodology for identifying the shearing and forming features from the nested layout and component respectively, with the help of illustrative examples, and real life components.

Methodology

With the above stated objectives, the *proposed methodology* for feature recognition is given below :

Initially pressworking operations are classified into two major categories, viz., *shearing and forming operations* based on the stresses applied during the pressworking process, and the type of features developed (i.e. 2–D or 3–D). These operations are further classified on the basis of their locations in the component and/or part layout. Based on the classification system of operations, pressworking features are classified into shearing and forming features, and these are also further classified, as in the case of operations, to identify them automatically from the CAD database.

CAD database information obtained for a component and/or layout created flexibly by a designer in wireframe modeling is considered as input to the system. The connectedness and entity groups (subgraphs) are determined by vertex fusion methodology, since the information about connectedness and subgraphs in a component graph are not available directly from the CAD database. With the help of plane equation determined by vector

normal method for a component, its nature (i.e. 2–D or 3–D type) is determined after validating the component graph with Euler's formula of wireframe model.

Shearing features are identified from the plan of 2-D component nested layout by neglecting the thickness of sheet metal. The entity groups are classified into raw material feature set, boundary feature set, and inside feature set by using set theoretic concepts, minimum enclosing rectangle and ray tracing technique, as features belonging to these sets are not obtained directly either from the CAD database or from the entity groups identified. Boundary feature and inside features belonging to a component are identified, and lexicographically ordered components are used to determine the type of layout and adjacent components. Adjacency component directional relationship (ACDR) table is determined by finding directions of components using cartesian octants and a set of rules developed to identify the adjacency relationship. Shearing features are then identified using different feature sets, layout, type of layout, and ACDR table. A set of principles have been developed by reasoning and relating entity groups geometrically and topologically for recognizing shearing features.

Forming features are identified using mean plane of a 3-D component. Initially, the component graph is processed to identify parent and child subgraphs by relating the subgraphs geometrically and topologically. Loops present in each subgraph along with the vertices and edges in a loop are identified with the help of plane equations and vertex fusion methodology. Curved planes and end planes are determined in a component, and adjacency plane relationship (APR) matrix is established. Cross-bend features are extracted from the subgraphs by removing common edges between the planes. Features are recognized by matching the feature graphs extracted from the component with the standard/pattern feature graphs developed using the principle of isomorphism of graphs.

In case of a formed component, a *flat pattern* is developed to recognize shearing features, and the louvering feature. This is because, shearing features are 2–D in nature and louvering feature is a combination of lancing (2–D feature) and bending (3–D feature). The flat pattern is developed using the principle of rotation about an arbitrary axis in space by taking thinning effect into consideration. Flat pattern is processed ensuring that its graph is in accordance to the definition of graph theory and Euler's formula for wireframe model, and louvering feature is identified. This flat pattern is taken as input to the CAD system and an optimum nested layout is created manually by wireframe modeler for identifying shearing features.

The proposed methodology is implemented in C language on PC/AT and HP 9000/850. The geometric data is obtained from the standard DXF output file of AutoCAD Release–10 as input to the system. The methodology is tested for both 2–D and 3–D components

with illustrative examples representing real life components.

Conclusions

The proposed methodology is a generalized one for identifying pressworking features from 2–D and 3–D components created by wireframe modeler.

The proposed classification systems help in determining features in a hierarchical manner, and thus reduce the search procedure during implementation. Also, the classification systems help in easy identification of finish operations by taking total tolerances into consideration.

The proposed methodology enables the development of alternative process plans for manufacturing a sheet metal component, since information about different operations by means of which the part can be produced from the raw material are identified. An optimum process plan can be selected from the generated alternative process plans based upon the requirements and facilities available in an organization. Thus, feature recognition of pressworking processes for a sheet metal component and/or layout is a step towards integrating the Computer Aided Design (CAD) and Computer Aided Manufacture (CAM). Hence, it helps in bridging the gap between CAD and CAM through Computer Aided Process Planning (CAPP) system, as it is the backbone of CAPP.

The geometrical information of identified features obtained from the system can be used in optimization of tool paths in case of CNC turret presses, off-line generation of G and M codes, mapping form features for the development of tool, for inspection of tools and parts, etc.

Chapter 1

Introduction and Literature Review

1.1 Introduction

Sheet metal components are widely used in various industries like aerospace, electronics, machine tools, refrigeration and air conditioning, etc., and they form a significant part of the manufacturing activity. Sheet metal components are important not only from functional point of view, but also from aesthetic point of view, since they are used as enclosures to cover the products and are visible to the outside world. These components vary in size, shape and in complexity, from being simple to very difficult to manufacture. Therefore, pressworking processes are receiving greater attention and are being widely applied by the metal working industries.

Manufacturing processes required for sheet metal components are identified by analyzing the drawing of the component/part layout, and thereby the design information is translated into manufacturing information. The manufacturing information is used in the development of process plans, generation of G and M codes for CNC turret presses, mapping form features in the development of tools for presses, inspection of press tools and components, codification of parts from manufacturing point of view, etc. At present, in most of the industries, the transformation of design information into manufacturing information and the subsequent utilization of manufacturing information for production of components, and in the other product cycle activities is done manually or interactively using computers. Due to the presence of this manual or interactive segment in the process cycle, the realization of a truly computer integrated manufacturing system (CIMS) (Fig. 1.1) is yet not possible in sheet metal industry.

In CIMS process cycle (Fig. 1.1), there are three segments viz., product and process definition, manufacturing planning and control, and factory automation, which take a product from the concept to the completed merchandise (Rehg, 1994). These three process segments share common data and resources as shown in Fig. 1.2 by overlapping areas.

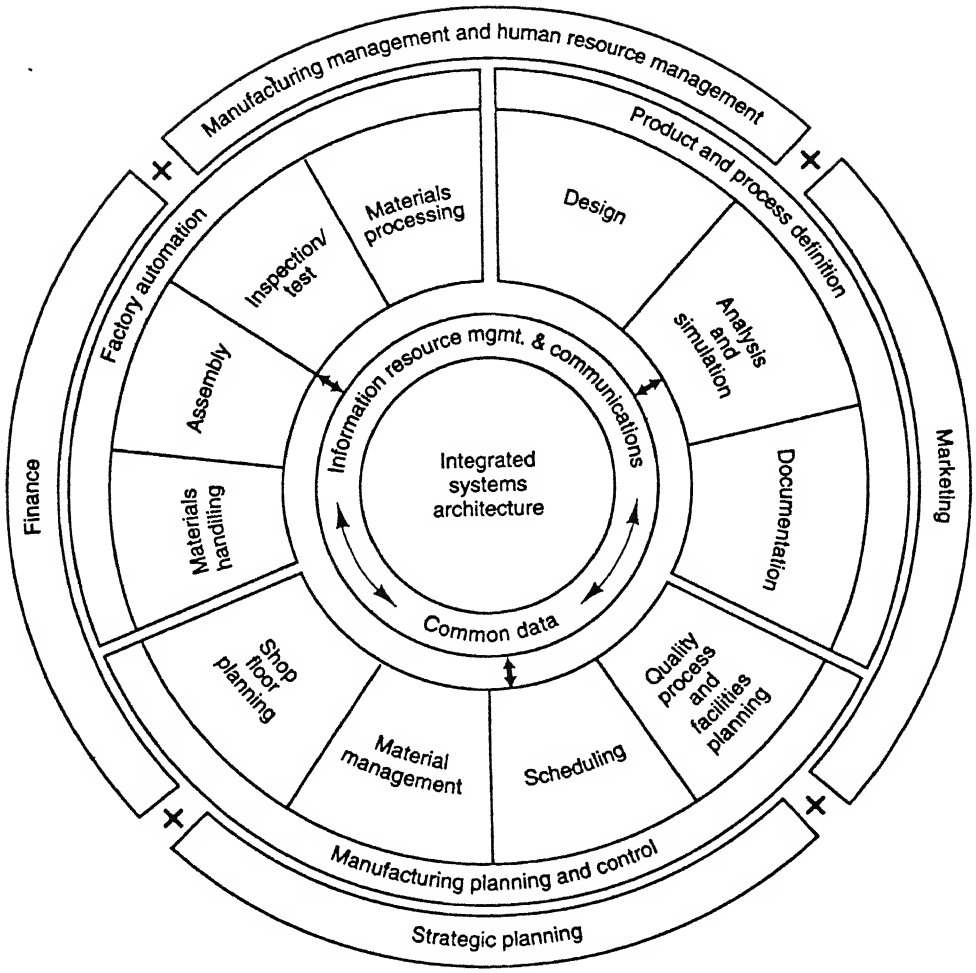


Figure 1.1: Society of Manufacturing Engineers CIM wheel (Rehg, 1994).

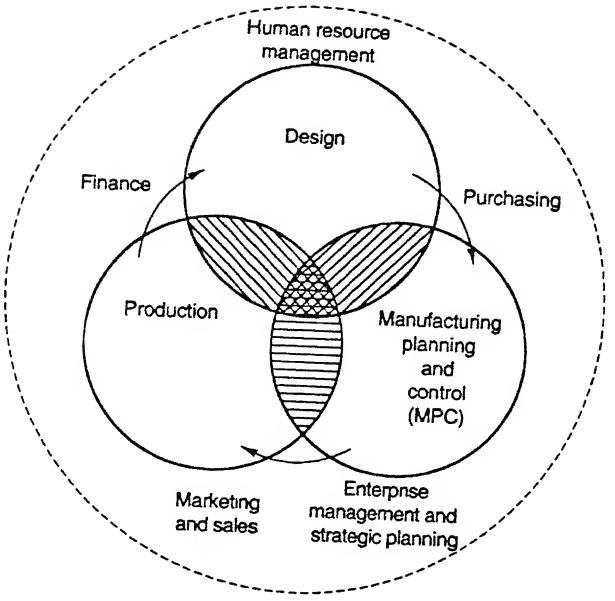


Figure 1.2: The enterprise areas (Rehg, 1994).

The shared data are usually about the design information of the product. To achieve the objectives of CIMS, the design information should be transformed directly and automatically into the manufacturing information. Therefore, the design information is created and stored in a computer aided design (CAD) database. The CAD database information has to be further transformed into computer aided manufacturing (CAM) information for manufacturing the components. These two aspects (i.e., CAD and CAM) of CIMS are its vital components, and are very well developed individually into two isolated islands in CIMS. These two islands are being connected by a bridge called computer aided process planning (CAPP). The basic component of a CAPP system is the transformation of design information into manufacturing information. The process of transformation of information (done either interactively or automatically) from design into manufacturing is called *feature recognition*, which essentially forms the backbone of CAPP. As such, automatic extraction of manufacturing information from CAD database helps in the integration of CAD and CAM, – a step towards realization of CIMS framework. Moreover, automatic feature recognition of components from the CAD database helps in eliminating the inherent problems associated with the manual or interactive system for transforming design information into manufacturing information.

In this chapter, initially the various schemes of representation of components in CAD database are given. It is followed by a literature review on feature recognition of components with special reference to sheet metal components. Subsequently, the scope and

objectives of the present research are given. The chapter concludes with an organization of the dissertation.

1.2 Representation of components in a CAD database

The drawing of a physical entity (component) is result of the design process of the component. It is the way of expressing the component in a communicable media. With the introduction of computers, the design information of components are represented usually by the following modeling processes :

1. Wireframe modeling,
2. Surface modeling,
3. Solid modeling, and
4. Feature based modeling.

It is very difficult to give a clear cut verdict about the best modeling technique since each modeling is meant for different engineering applications. The requirements of each application are different. There are two basic levels of information required for any engineering application (Zeid, 1991). They are geometric information about the component, and attributes for the geometry from engineering analysis point of view. In the present work, the CAD database of a wireframe model is considered from the feature recognition point of view such that manufacturing features can be recognized. With this as the objective, the above four geometric modeling techniques are discussed briefly in the following subsections.

1.2.1 Wireframe modeling

The wireframe model of an object looks like a stick figure or an edge representation. Fig. 1.3(a) shows a wireframe model of a 3–D prismatic component. Figs. 1.3(b) and 1.3(c) show a 2–D nested layout and a 3–D sheet metal component respectively. The term *wireframe* is related to the fact that one can imagine a wire that is bent to follow the object edges to generate the model. This is the simplest geometric modeling technique among all others mentioned above. On the other hand, it is the most verbose geometric model used to represent the component. The primitives used in wireframe modeling are points, lines, arcs, circles, ellipses, parabolas and synthetic curves.

The database of a wireframe model is simply an itemized collection of all the vertices (points of 2–D or 3–D coordinates) and edges (straight lines and 3–D curves) present in

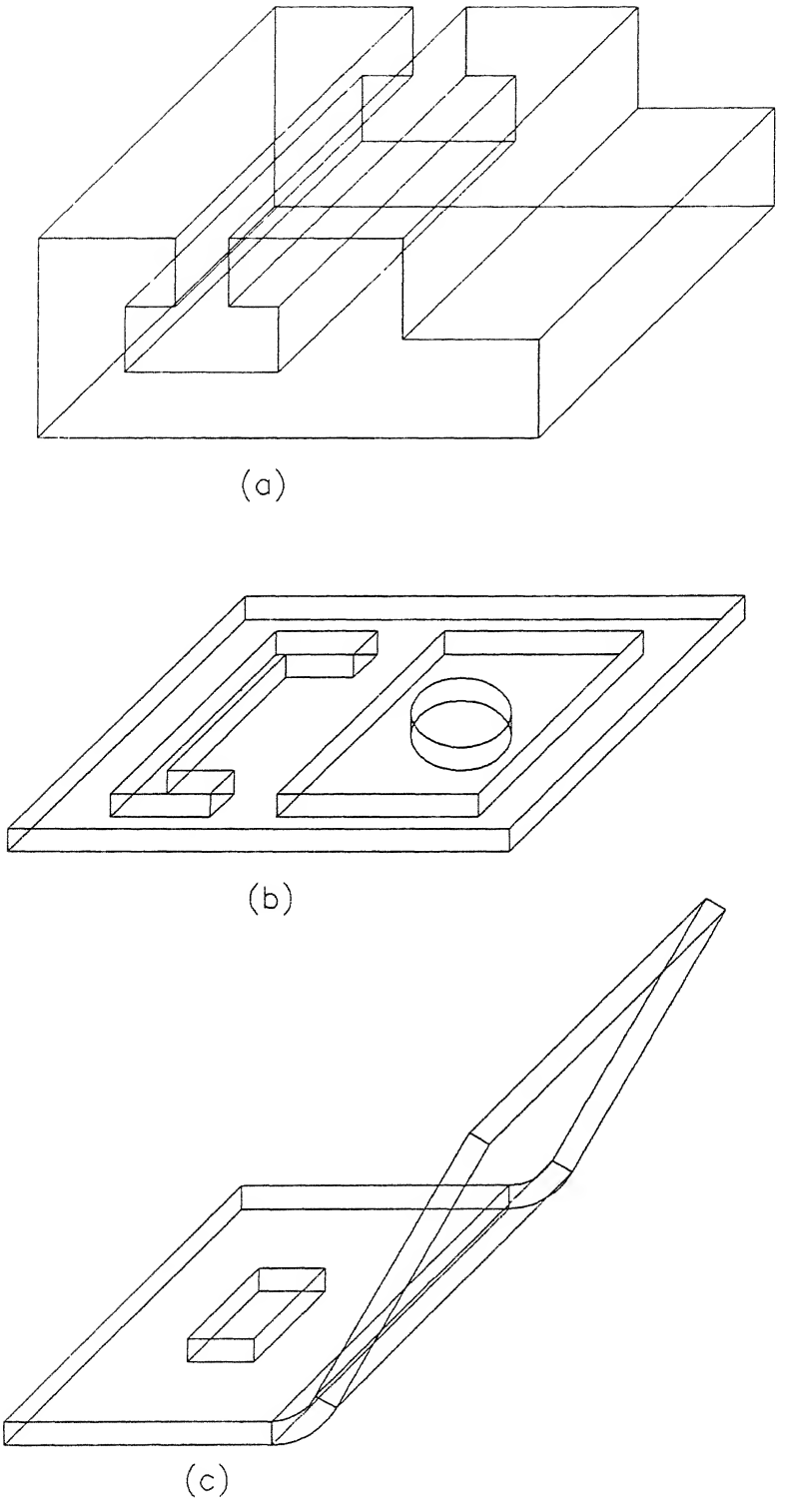


Figure 1.3: Wireframe models : (a) prismatic, (b) 2–D sheet metal part nested layout, and (c) 3–D sheet metal component.

an object. Therefore, the data structure does not give the connectivity of edges in an object (Fig. 1.4). Also, the wireframe model has two serious shortcomings (Agarwal and Waggenspack Jr., 1992) viz., inability to distinguish what is the *inside* or what is the *outside* of an object (Fig. 1.5), and the cognitive ambiguity (Fig. 1.6). From Fig. 1.5, it can be seen that, decision about the front and the back sides of the cube is difficult to make without further information about the entities. Also, if a person concentrates more on the picture, it looks as if the front and back sides swap back and forth. In case of the most commonly used tesserac (Fig. 1.6(a)), without any additional information about the surfaces, it is difficult to find the true model. As shown in Figs. 1.6(b) to 1.6(d) all are the valid wireframe models.

1.2.2 Surface modeling

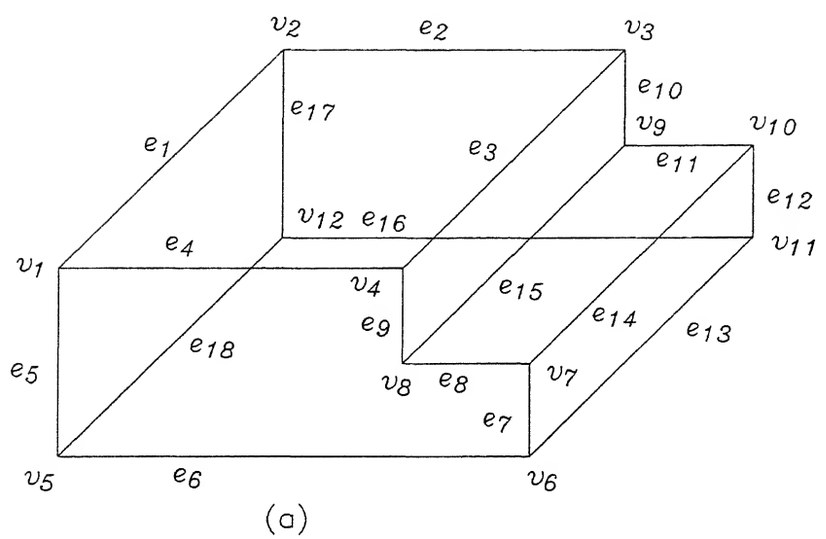
Wireframe model cannot be used to represent complex objects like airplane bodies, turbine blades, automobile bodies, etc. In this type of applications, surface modeling is commonly used to represent a part. The exploded view of a simple object showing its boundary faces is depicted in Fig. 1.7.

In most of the CAD packages, surface models for components are created using the wireframe model as the starting point. The primitives used in this model are plane surfaces, ruled (lofted) surfaces, surface of revolution, tabulated surfaces, bezier surfaces, coons surfaces, etc.

Surface modeler gives the detailed surface information with the explicit representation of the wireframe data. Surfaces of an object represent the primary interface (or boundary) between the points, and help to determine vertices inside the volume and those outside. Thus, surface models are more complete when compared to wireframe models. The data structure for surface modeling of a stepped cube (Fig. 1.8(a)) is given in Fig. 1.8(b). In the data structure of this model the connectivity of edges is available directly from the CAD database, since the surface is made up of edges and vertices. The model still does not represent the object completely since it represents only the geometry of the object. The connectivity of surfaces cannot be obtained directly. As such, the connectivity of surfaces has to be determined from the data structure.

1.2.3 Solid modeling

In this branch of geometric modeling, emphasis is given to the creation of complete objects such that it is adequate for answering arbitrary geometric enquiries algorithmically without interaction of human beings. A solid model defines the complete geometry and topology of an object. Therefore, given the solid model of an object, it is possible to determine



e_1, v_1 & v_2	e_2, v_2 & v_3	e_3, v_3 & v_4	. . . .	e_{17}, v_{12} & v_2	e_{18}, v_5 & v_{12}

(b)

Figure 1.4: Data structure of a wireframe model : (a) Stepped cube, and (b) Data structure of the stepped cube giving edge information.

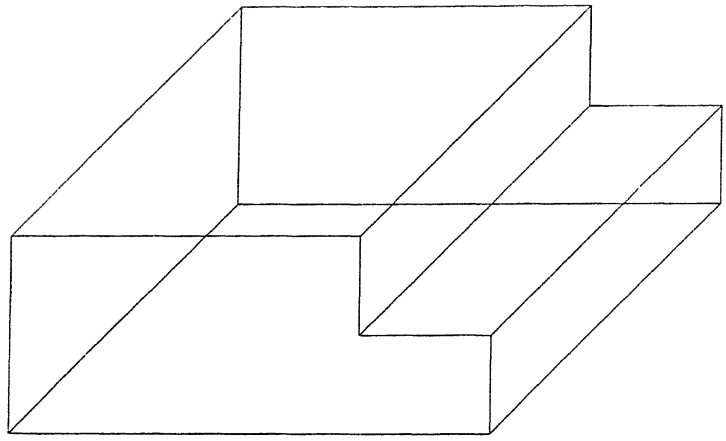


Figure 1.5: Wireframe model for a stepped cube.

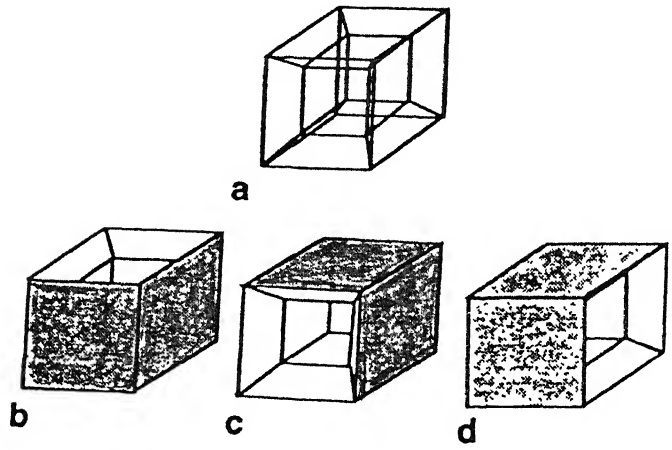


Figure 1.6: Ambiguous wireframe models : (a) Tesserac, (b) (c) and (d) valid planar-faced solids (Agarwal and Waggenspack Jr., 1992).

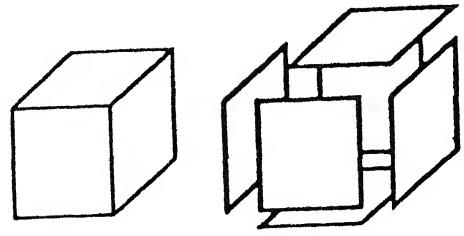


Figure 1.7: Boundary faces of a cube(Agarwal and Waggenspack Jr., 1992).

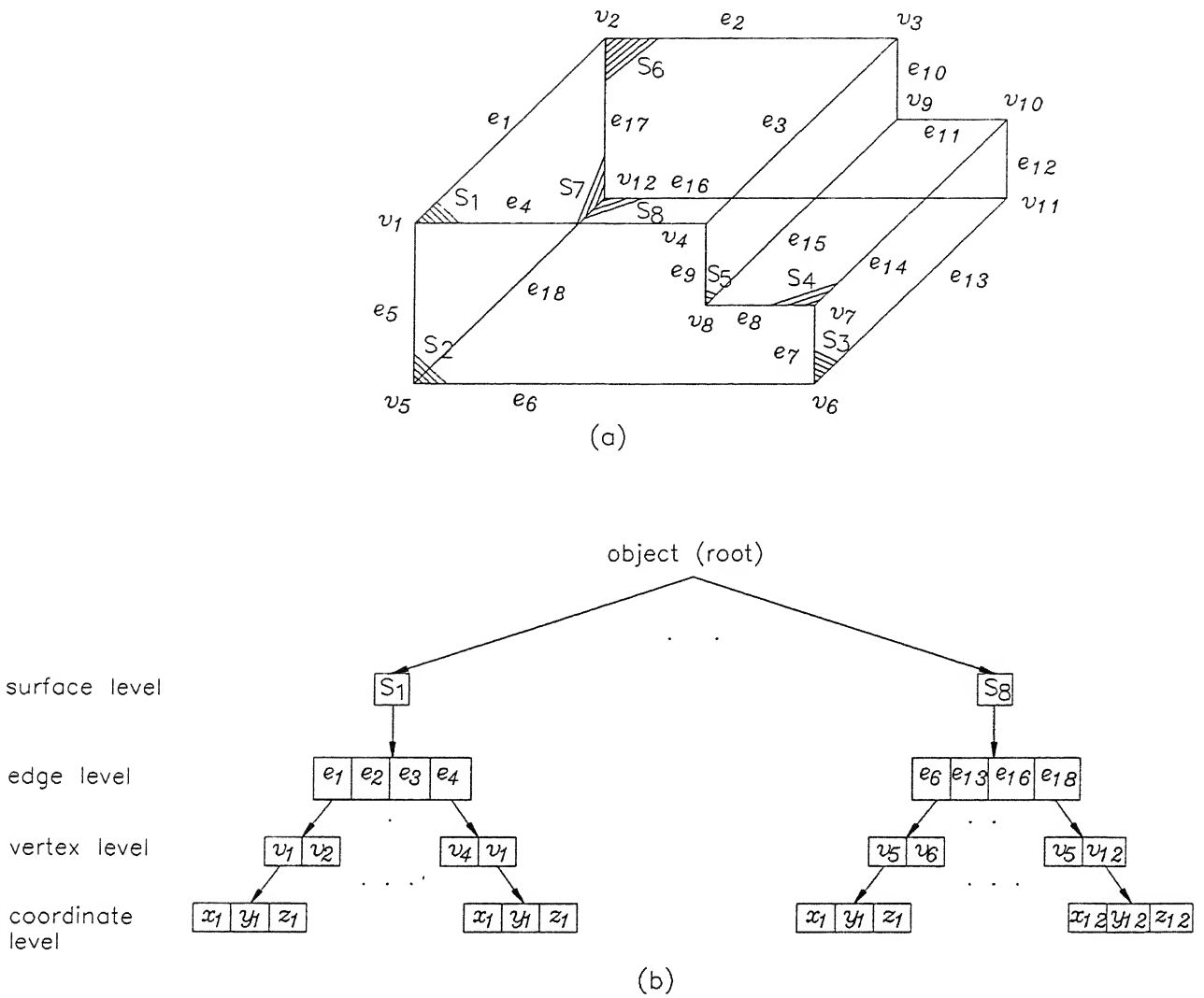


Figure 1.8: Data structure of a surface model : (a) Stepped cube, and (b) Data structure of the stepped cube.

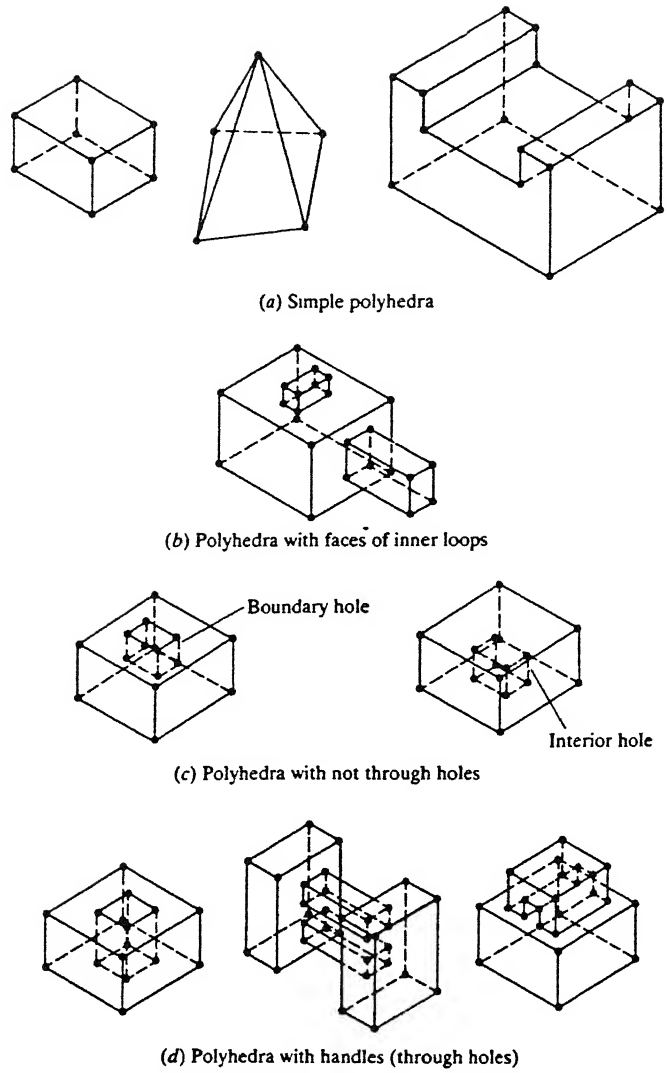


Figure 1.9: Types of polyhedral objects (Zeid, 1991).

whether the point under consideration is inside, on the surface (i.e., on the boundary) or outside the object. This classification of the point is termed as *spatial addressability* (Zeid, 1991).

The topological information of solid modeling helps in determining the connectivity of surfaces in an object. Based on the internal representation scheme, different types of solid systems are identified. Most of the solid modeling systems have two types of internal representations, viz., boundary representation (B_rep) and constructive solid geometry (CSG). The primitives used in B_rep are faces, edges and vertices, and the objects are classified as either polyhedral or curved type. Various types of polyhedral objects are shown in Fig. 1.9. General data structure for a boundary modeling is given in Fig. 1.10.

The primitives used in a CSG model for the construction of an object are blocks,

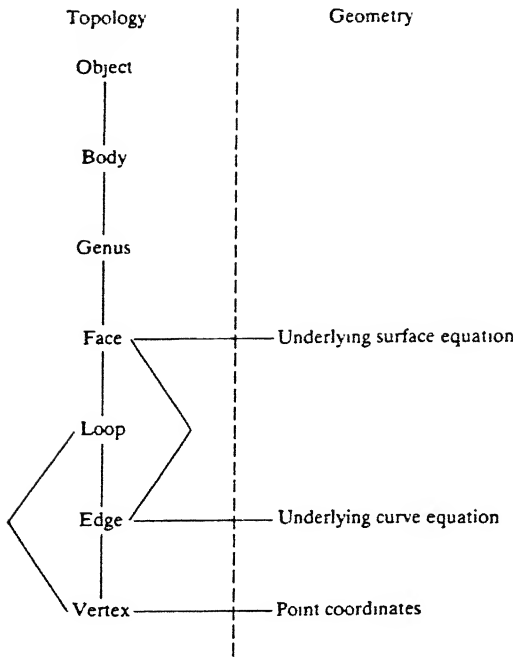


Figure 1.10: General data structure for a boundary modeling (Zeid, 1991).

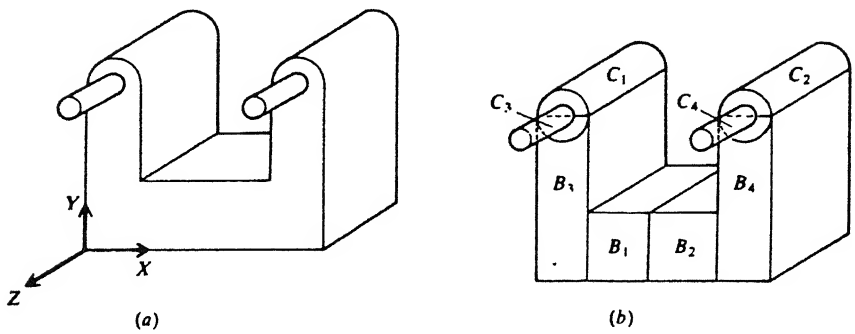


Figure 1.11: A typical solid and its building primitives : (a) Typical solid, and (b) Primitives (Zeid, 1991).

cones, spheres, cylinders, torous, etc. In almost all CAD packages, CSG is used as the core technique for constructing objects and depending upon the requirements CSG can be converted into B_rep or to other representation schemes. A CSG model for an object is given in Fig. 1.11 and its CSG graph is shown in Fig. 1.12. Data structure of a CSG for the above object is given in Fig. 1.13.

The information stored for representing a component by the solid modeler is much more than the information required for wireframe modeling and surface modeling, since it requires both geometrical and topological information of the part. Solid models cannot automatically create other models from their definitions and they cannot use the data created in other models to create a solid automatically.

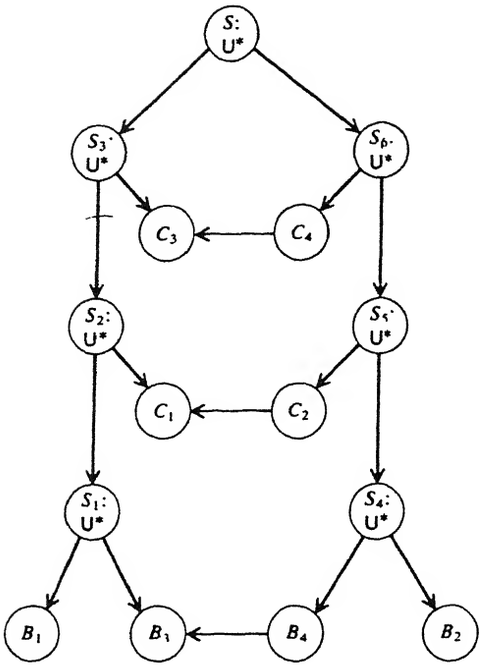


Figure 1.12: CSG graph of a typical solid (Zeid, 1991).

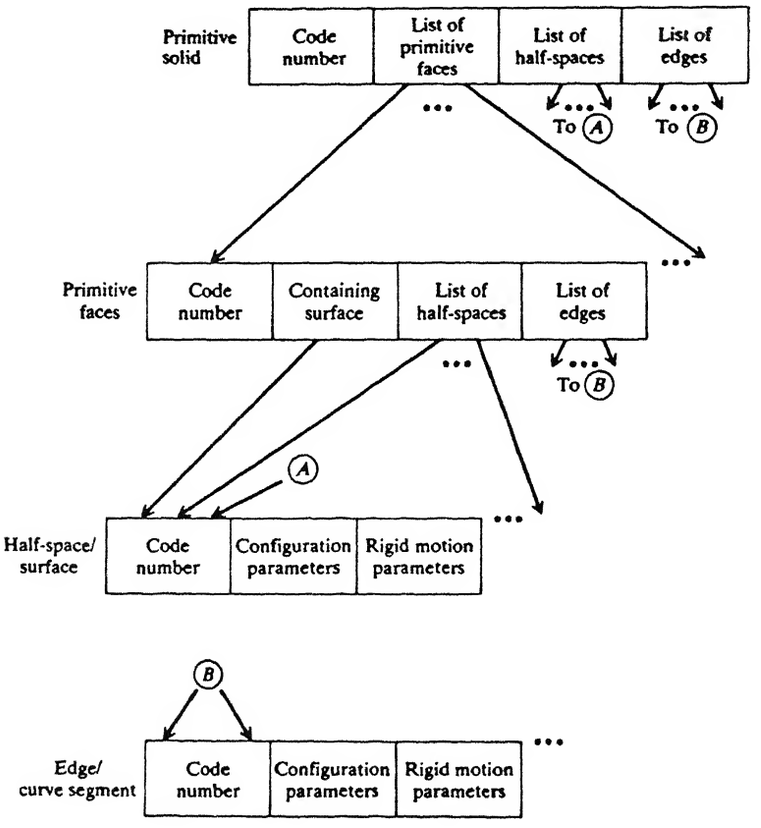


Figure 1.13: Data structure of a typical solid Primitives (Zeid, 1991).

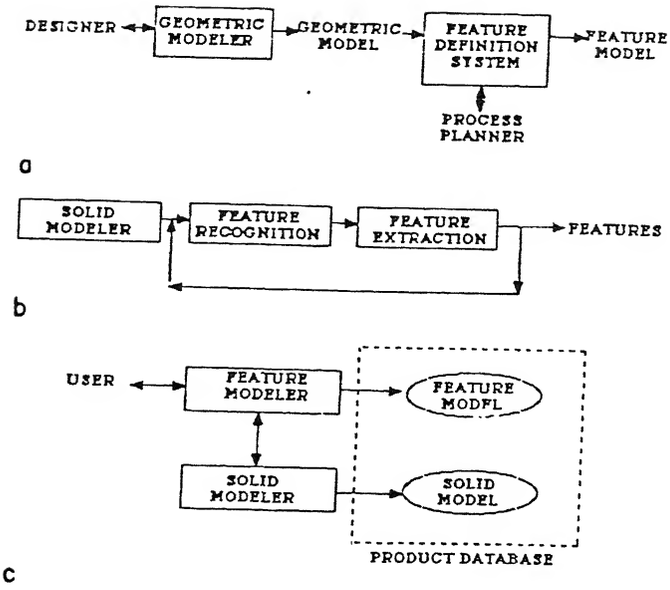


Figure 1.14: Schematics of feature-definition approaches : (a) Interactive features definition, (b) Automatic feature definition, and (c) Design by features (Shah, 1991).

1.2.4 Feature based modeling

In feature based modeling, objects are created by features. Each feature is represented by a specific geometry and feature attributes. The attributes of features can be dimensions, dimensional tolerances, manufacturing notes etc. Thus, various types of information are included depending on the application requirements. It should be noted that features are independent of geometry. Feature models are created by three approaches viz., interactive feature definition, automatic feature definition, and design by features (Shah, 1991). These concepts are illustrated in Fig. 1.14.

Shah and Rogers (1988) have used a four level data structure to represent the object created in feature based modeling. For an abstract object, the data structure is shown in Fig. 1.15. In the figure, the top level records the feature relationships, second level consists of generic properties of each feature, third level stores instance data of the feature, and the fourth level records the CSG tree of the object.

It is expected that the feature based design will provide a better approach to integrate design with other applications where the design information is required. But, at present, the major disadvantage associated with it is that each application of engineering requires a different set of features. Also, in this modeling, still many issues like feature representation, feature validation, features and constraints, multiple views on features, features and languages, features and standardization are not yet resolved completely (Solomons, et al., 1993).

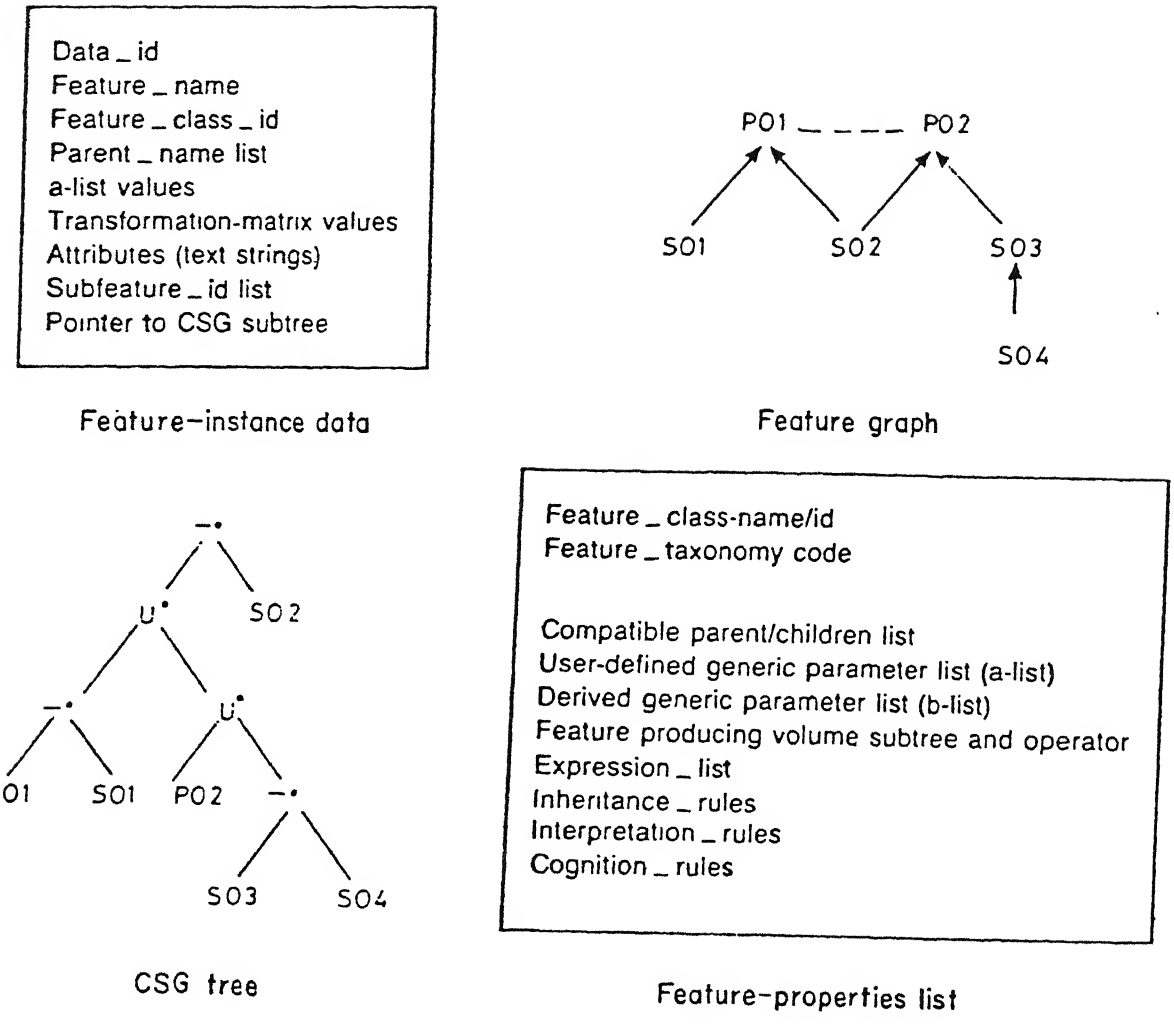


Figure 1.15: 4-level data structure of a feature model (Shah and Rogers, 1988).

In the above section, the various means of representation of objects by the geometric modeling techniques are explained briefly. In the following section, a l review of the various approaches used to recognize manufacturing features from t database are discussed.

1.3 Literature review

In the literature, we find that so far there is no clear and exact definition for t *feature*. Oxford dictionary defines feature as : (1) the parts of the face which form its appearance, (2) distinctive characteristic; aspect, and (3) special or pr article or program. In case of Webster dictionary, feature is defined as : (1) the st form, or appearance, especially of a person, (2) the makeup or appearance of the fa parts, (3) prominent part or characteristic, and (4) a special attraction. Therefo engineering point of view, features can be viewed as information sets that refer tc aspects of form or other attributes of a part.

In engineering, the product cycle has three distinct phases viz., conceptual desig design phase, and manufacturing phase (Lenau and Mu, 1991). The same feat defined differently in each phase. In the *conceptual design phase*, features are from the *functional point of view*, and are called *functional features*. The same in the *design phase* are defined from *shape aspect*, and are called *form features*. *manufacturing phase* same features are defined from the *production point of view*, called as *manufacturing features*.

For example, consider the sheet metal component (Fig.1.16) used as a motc The motor cover requires some space for aligning the motor shaft and for aligni to fix the cover on the casing of the motor. In the conceptual phase, these feat defined from the functional point of view, and are called shaft and screw alignment These features in the design phase are defined from shape point of view, and they a circle and obround features. In pressworking (i.e. manufacturing phase), these feat defined from production point of view, and are called piercing features. Thus, t feature of the component has different feature names in different phases of the cycle.

1.3.1 Feature recognition in general

In geometric modeling, parts are represented by geometric entities.Therefore, feat defined as a group of geometric entities, which put together will give some higl meaning. The geometric entities representing form features of a component are e:

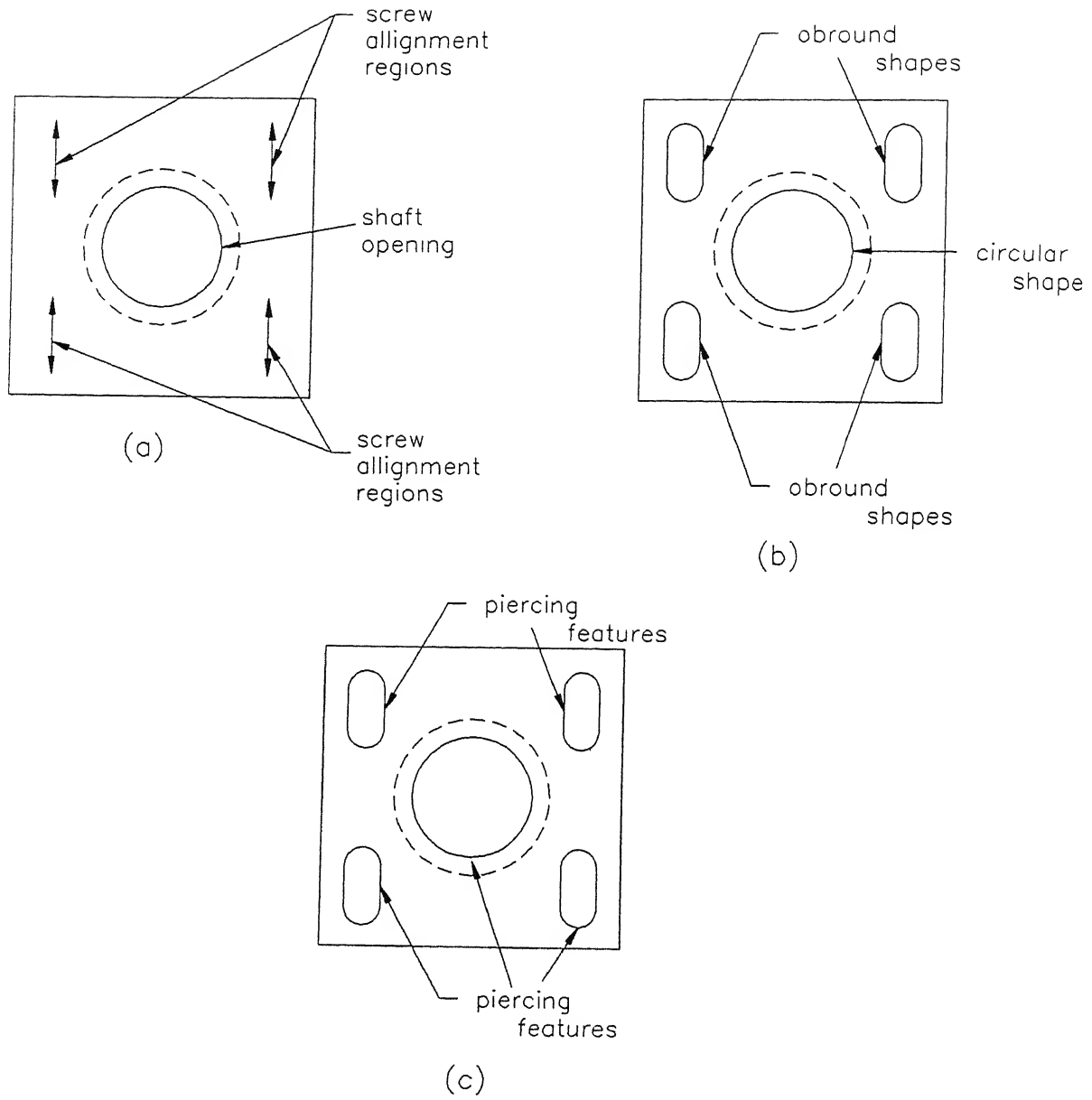


Figure 1.16: Various types of features : (a) Functional features, (b) Form features, and (c) Manufacturing features.

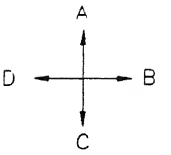


Figure 1.17: Pattern primitives.

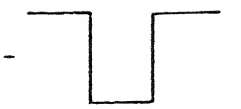


Figure 1.18 A simple hole drawing.

from the CAD database, and manufacturing features are recognized. The process of extraction and recognition of features is called *feature recognition process.* Generally, features are recognized through the following five major approaches, which have proved to be significant (Chang, 1990). They are :

a. Syntactic pattern recognition,

b. State transition diagram and automata,

c. Decomposition approach,

d. Logic approach, and

e. Graph based approach.

Syntactic pattern recognition process consists of defining the primitives in terms of pattern grammars. Fig. 1.17 gives the pattern primitives, and $BBCCCBBAAABB$ gives the string developed for a simple hole (Fig. 1.18), assuming that the primitives represent one unit length. The picture of the object is represented by the concatenation of pattern primitives and a parser is used to identify whether the given picture satisfies a certain grammar or not. An architecture for a syntactic pattern recognition based recognizer is given in Fig. 1.19. This approach is most suitable for 2–D and $2\frac{1}{2}$–D components, since the pattern grammars can be developed easily by following a fixed order to represent the primitives. In case of 3–D components, pattern grammar development is tough, since following a fixed order to represent a primitive is difficult.

Kakino (1977) has developed a part description method on the basis of the fundamental concept of converting the part drawing into computer oriented information for the data structure. Based on his work, Jakubowski (1982) used syntactic pattern to describe 2–D profile information of $2\frac{1}{2}$–D components created by sweeping. Features are recognized based on the occurrence of string of primitives. He applied extended free grammar to describe machined part families and gave a detailed explanation for parser construction.

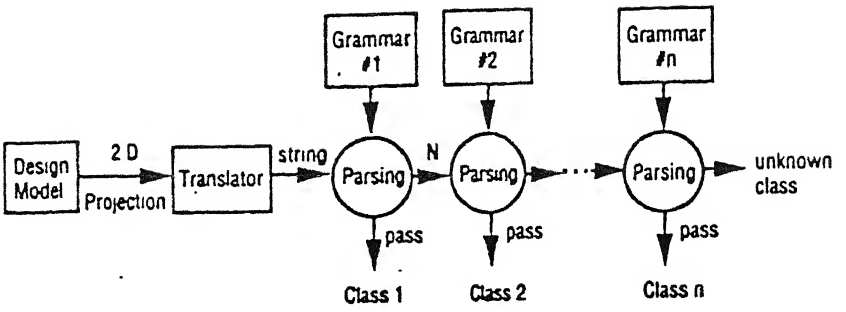


Figure 1.19: An architecture for a syntactic pattern recognition based feature recognizer.

Kryprianou (1980) (source Chang, 1990) has used B_rep database to extract information concerning types of features present in an object. He identified depressions (holes) and protrusions (bosses) in the prismatic parts. A rule based system has been used to identify GT codes for parts. In this system, initially a face set data structure is constructed using a feature recognizer. Afterwards using feature grammar, the data is parsed and features are recognized. Srinivasan and Liu (1985) have proposed definitions for shapes (symmetric in nature) produced by machining centers using pattern grammars. Thus, machining processes are identified directly for 2–D components.

Li (1988) used syntactic pattern recognition method to identify turning features for axi-symmetric components from their 2–D profiles. Pattern primitives are predefined, and pattern string is developed by travelling the 2–D profile from left to right. This pattern string is compared with the predefined pattern string to recognize the turning features. In this system, each feature is required to be pre-defined initially. Features present on the slant surfaces of a component cannot be identified. Sahay et al., (1990) have developed a system to identify the features present on the slant surfaces from the 2–D profile of the axi-symmetric rotational components. Initially, types of faces present in the profile are identified along-with information about the adjacent faces. Then the features are classified into simple and complex features. These features are identified by comparing with the predefined primitive turning features.

State transition diagram and automata approach for feature identification is similar to that of syntactic pattern recognition process. Here the part geometry is described by sweeping operations and/or the union of sweeping volumes. The generating surface is described by ordered pattern primitives, along with technological information and approach direction. The machining feature on the generating surface is recognized using a state transition diagram. For a step and slot, the state transition diagram is shown in Fig. 1.20. In this approach, instead of using grammars and primitives, relationships of adjacent pattern primitives are used by assigning a value of zero to convex adjacency and one for

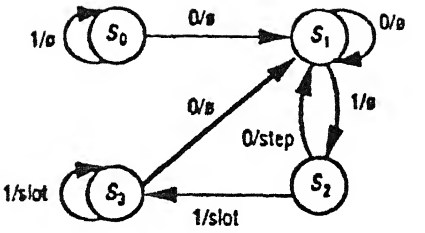


Figure 1.20: State transition diagram for step and slot (Chang, 1990).

concave adjacency. Within the class, classification is done by feasible approach. Iwata et al., (1980) used this transition approach to identify features in their CIMS/PRO system. Milacic (1985) used rules for developing grammars for members of the part family, and composite rules for the part family by combining the rules of part family members. These composite rules are called *automata*. Given the grammar of the part, its classification is done by testing it against the composite rules.

The above two approaches are meant only for simple parts, and grammar for the primitives should be defined before these approaches are employed to recognize features.

Decomposition approach decomposes the geometric model of an object into several smaller volumes to recognize features. Woo (1982) has developed the idea of alternative sum of volumes (ASV) technique to represent an object by a series of convex components with alternating signs (for volume addition and subtraction). ASV represents feature primitives of the component, and is used for $2\frac{1}{2}$–D components developed by B_rep. Decomposition approach is given in Fig. 1.21. In this, the raw material is considered as a convex hull C, and the difference between the convex hull and the part P is called H. In the next iteration, P is replaced by H, and C is determined for this H and the process of finding H is repeated until H is a null set, or a cycle is detected.

Ferreira and Hinduja (1990) have used convex hull based feature recognition method for $2\frac{1}{2}$–D components created by B_rep solid modeler. Convex hull is determined for the object, and edges within the convex hull belonging to faces forming a feature are determined along-with edges of inner loops and concave edges. Features having common geometric characteristics are merged, and then manufacturing features are identified. Features are identified by considering one face at a time rather than the entire component. The approach direction to the part is also identified. Kang and Woo (1991, parts 1 and 2) have given algorithm aspects in their part-1 paper, while the part-2 paper deals with the cause of the non-convergence of ASV process. The complexity of the proposed algorithms has also been discussed in the papers.

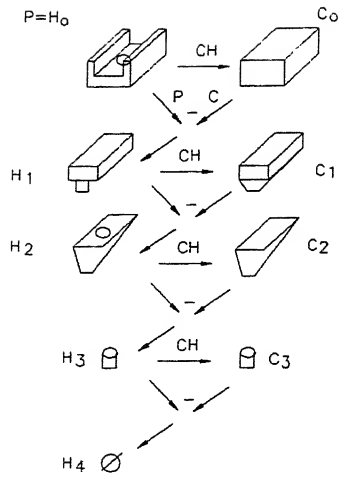


Figure 1.21: Derivation of ASV series.

Decomposition approaches have been used in the generation of NC cutter path (Okino and Kubo, 1970., Armstrong, et al., 1984), and in the generation of face topologies for components created in wireframe modeling (Agarwal and Waggenspack Jr., 1992). The major disadvantage of the decomposition approach is that it may produce smaller features which are unrealistic in nature from the manufacturing point of view. Therefore separate feature recognition is required to identify the manufacturing features.

Logic approach is associated with an expert system. In this method, the topological structure of features is written into logical rules. The components are usually created by B_rep geometry models. Henderson and Anderson (1984) developed a production rule by means of which the topological and geometrical information of the manufacturing features are captured. The production rules are in the form given below :

$$\text{If } C_1, C_2, C_3, ..., \text{ then } A.$$

where C_1, C_2, C_3, ..., are conditions, and A is the manufacturing feature. Production rules are simple to write for 2–D and $2\frac{1}{2}$–D components, but difficult for 3–D components. From the search point of view of features, logic approach is inefficient and computationally intensive. Also, development of rules for intersecting features are quite difficult. Languages used in this system are Prolog and Lisp.

Wang and Wysk (1988) have extracted features from a CAD database for axi-symmetric objects for the automated process planning (APP) system. For extracting geometric entities from the CAD database, initially certain rules are developed. These entities are then

determined by another set of rules. From form features, the object is reconstructed and then manufacturing features are identified. Details of the algorithm and heuristics developed are not given. van't Erve and Kals (1986) have used an interactive system to define a part geometry. Manufacturing features are identified in their XPLANE–CAPP system.

Meeran and Pratt (1993) have used Prolog language to identify manufacturing features of 3–D prismatic parts created in wireframe modeling. The orthographic views of a component are considered for feature recognition. Simple features from simple objects are identified. Features are extracted from the CAD database and compared with the primitive features defined earlier by rules. Once a feature is identified, it is deleted from the feature list and the process of identifying features is continued, till all the features are identified.

Graph based approach along with rules was used by Joshi and Chang (1987, 1988), and Joshi et al., (1988) to identify manufacturing features for components created in B_rep geometric modeler. An attributed adjacency graph (AAG) for the part is determined. An AAG consists of a set of nodes (N), a set of arcs (A), and a set of attributes (T). Thus AAG is represented as $G = (N, A, T)$. Each face is represented as nodes, and the adjacency between them is represented by arcs. If adjacent faces make concave angle, then the attribute of the arc is represented by 0, and if it is convex, then it is represented by 1. An AAG of a part is shown in Fig. 1.22, and some feature instances are shown in Fig. 1.23.

Woodwark (1988) has provided the following ideas about the use of set theoretic approach for recognizing features from CSG model of 3–D components.

1. To restrict the domain of the model, that is the range of primitives and/or of orientations that they may assume.

2. To restrict the ways in which primitives may interact spatially.

3. To restrict the allowable forms of the set theoretic expression that defines the model.

Marefat et al., (1990), and Marefat and Kashyap (1990) have extracted shape features from the B_rep polyhedral objects. They have used *cavity graphs*, which are developed to extract depressions of the intersecting features. Cavity graphs are extension of AAG, where in along with the explicit representation of convexity and concavity of faces, they represent explicitly the spatial orientation of faces relative to one another in 3–D space. This approach requires correct labelling of nodes, and features are searched to recognize from single cavity graph. Therefore, if features have several disconnected cavity subgraphs, then they are not recognized, as search is not carried out across the cavity subgraphs.

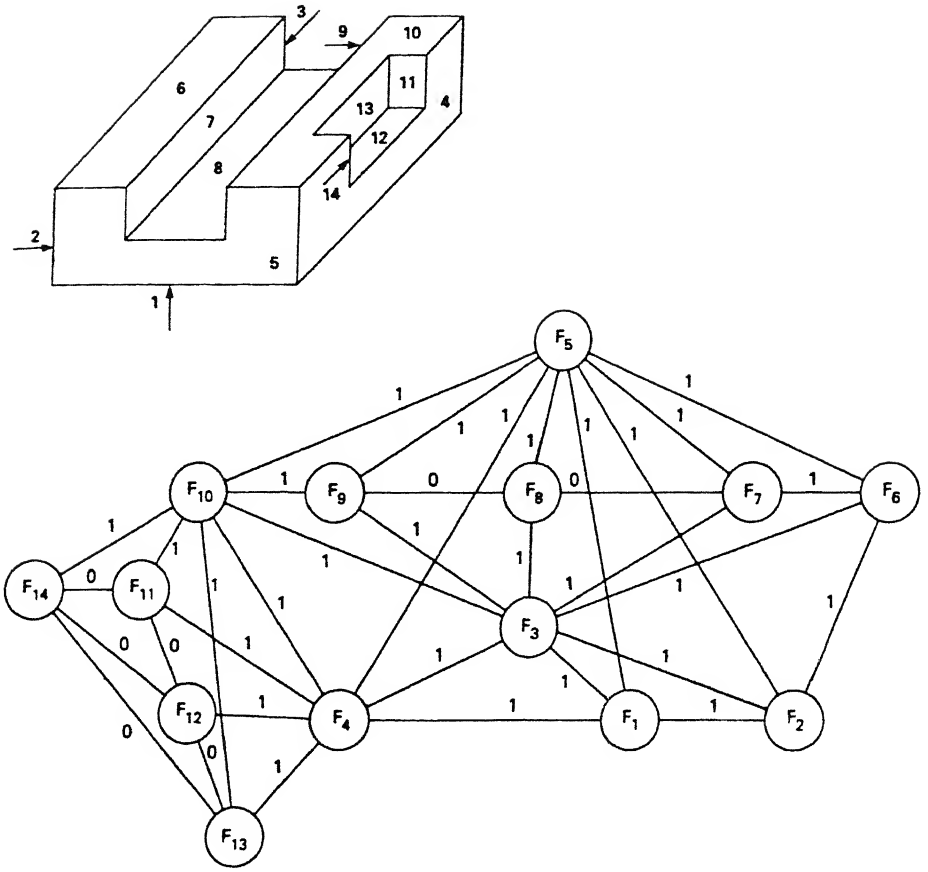


Figure 1.22: Example of AAG for a part (Joshi and chang, 1988).

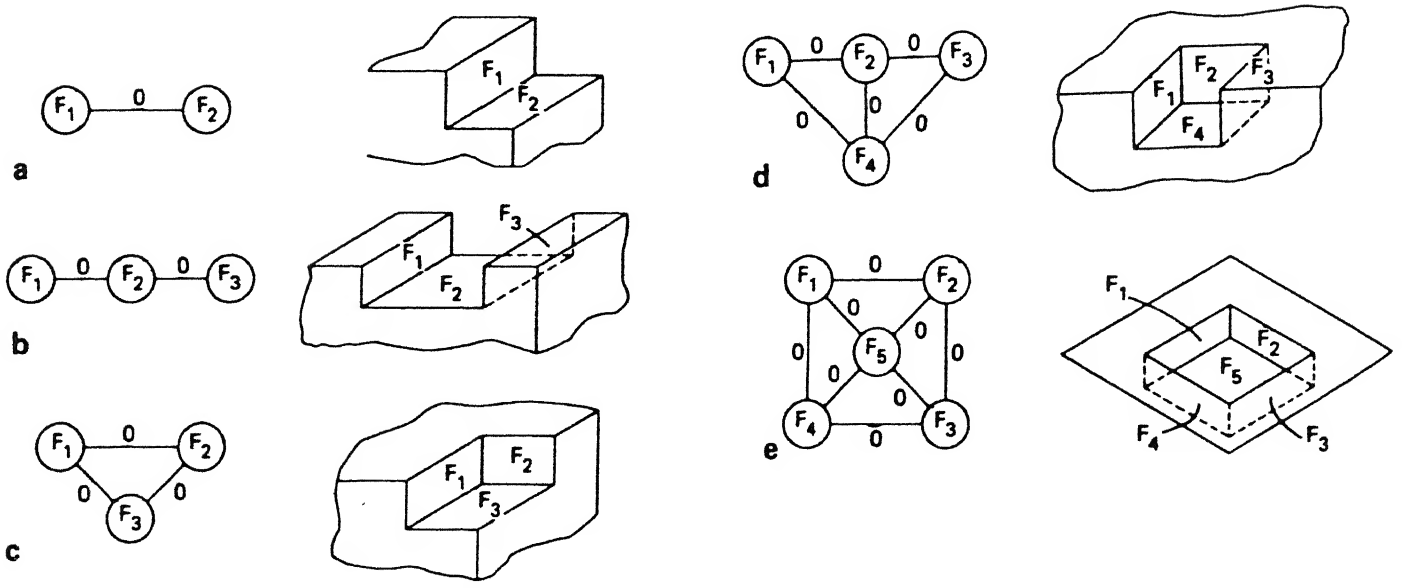


Figure 1.23: AAG of feature instances (Joshi et al., 1988).

By using the destructive solid geometry (DSG) tree of a component, Gavankar and Henderson (1990) have identified protrusions and depressions. The component has been created by B-rep models, and their topological properties are used to extract features from the model. This approach is not suitable for intersecting features, and features that can be approached from more than one face of the component. Also, the algorithm is dependent on the type of data structure used in the CAD database.

Shpitalni and Fisher (1991) have used quadtree encoding, and pruning the CSG tree for recognizing machining features for 3–D components created by CSG geometric modeler. In this approach, disjoint machining region comprising the total volume to be machined first is extracted, and then each disjointed machining region is dealt with independently. Machining regions are extracted by manipulating the CSG tree representing the component, and thus compact trees of CSG, where each compact tree representing a machining region is obtained.

Corney and Clark (1991) have recognized holes and pockets connecting multiple faces in $2\frac{1}{2}$–D objects using face-edge graph (FEG) for B_rep solid modeling. FEG are very large and complicated graphs, and are processed further to identify aspect face-edge graph (AFEG). AFEG is a subgraph of FEG after deleting nodes (also arcs incident on them), and their normal vector is either parallel or anti parallel to the direction of the vector from which the object has to be viewed or approached. This vector is called aspect vector. AFEG is processed further to identify machining features. The complexity of the algorithm in the worst case analysis is $n!$, where n is the number of vertices in FEG. The algorithm is suitable only for simple holes and the faces of the holes should be parallel to the aspect vector. Also, intersecting features and holes having undercuts cannot be identified.

Gavankar (1993) has used connectivity class in edge-face graph of a part of 3–D solid model to extract features like holes, protrusions and depressions. These features are classified and identified based on the topological and geometrical information obtained from the CAD database of morphological features (like chamfers, protrusions and depressions). This approach is used for prismatic parts having simple features, and the approach direction to the feature should be either of one or two faces only. It also requires data structure of multiple edge loops, and features are extracted after the faceted boundary models are converted into their exact form.

Ganter and Skoglund (1993) have used a B_rep solid model to identify features concerned with cores of casting. The cores of internal voids, single and multiple surface holes and boundary perturbations are identified using B_rep and graph based approach.

Chaung and Henderson (1990) have used vertex classification and vertex-edge (V-E) graphs for identification of features present in components represented in 3–D by B_rep

solid modeler. Vertices of the solid model are classified by analyzing the topology and geometric properties surrounding the vertices. The nodes of the V-E graphs representing the vertices are labelled in accordance to their type. The graphs of the shape primitive patterns are labelled and these graphs representing single regional patterns are matched with the pattern graph and subgraph of the component to recognize the features. The disadvantage of this approach is that there is no one-to-one mapping between VE graph and its corresponding regional graph representing morphological information. To apply this approach, convexity and concavity of edges connecting vertices of the B_rep model should be sorted in the database.

1.3.2 Feature recognition for sheet metal components

In the previous section, the current literature review of feature recognition for rotational and prismatic parts is given, and in this section, state of the art of the feature recognition for sheet metal components is given. Features are recognized from the CAD database obtained from the following geometric modelers :

1. Wireframe modeler,
2. B_rep solid modeler, and
3. CSG modeler.

Also, in few cases feature based design and structural description of sheet metal components are used for feature recognition.

Tulkoff (1981) (Source, Chang and Wysk, 1985) has developed a variant process planning system for sheet metal components of Lockheed. Ehrismann and Reissner (1988) have developed AIS-CAD system. This is a rule based expert system, which suggests the most economical way of manufacturing a flat sheet part, and the system generates NC codes. The processes considered are stamping, laser cutting, or combinations of both stamping and laser cutting. However, these papers do not give any detailed information regarding the approaches used to identify the features required for process planning.

Raggenbass and Reissner (1989, 1991) have used an expert system approach for recognizing the geometry of a part for automatic generation of a complete NC production plan for stamping, laser cutting, or for the combination of both the processes. The part is created by a wireframe modeler. The contours of the part are subdivided into inner contours and edges. It distinguishes between known (i.e., well defined) shapes (like circles, squares, rectangles and extended holes) with special attributes and unknown (i.e., not well defined or irregular) shapes. The decision module of the system selects one of the manufacturing

variant, i.e., complete stamping, laser cutting with movable table, laser cutting with movable head, and the combination of stamping with laser cutting. The selection is dependent on the complexity of the part geometry, the number and complexity of the inner contours, the number of parts to be manufactured, the number of parts to be produced from a sheet, required dimensional accuracy, surface quality and the thickness of the sheet metal. Tools are selected for known shapes by rule-based techniques, such that the required contour can be obtained in one or two blows of the punch. For unknown shapes requiring more than two blows, the tools are selected on the *collision consideration.* A laser operation is selected for the entire contour if it has arcs which cannot be produced with the available tools. The system considers only two manufacturing processes. Also, only 2–D layouts of similar components are considered for recognizing features.

Nnaji et al., (1991), and Kang and Nnaji (1993) have given a set of principles for extracting and recognizing features from sheet metal parts created by solid modeling using graph theoretic concepts. The component is represented by a 3–D solid model, and features are recognized by modifying the boundary information. The entities are grouped and the sheet is flattened. The features are classified based on their face orientation into two main categories viz., passage and non-passage, and are used as the basis for recognizing pressworking features. The criteria used are Euler formula, topological configuration, and angle between the feature faces of the components. A set of principles is developed by studying the face oriented representation of features, and concepts are derived by relating the geometrical form with the manufacturing process. All the features of sheet metal pressworking operations are not identifiable. Also, the features are identified from a single component, after converting the 3–D solid model into a 3–D face oriented representation.

Hoffmann et al., (1992) have developed a system called MINICAP (a CAP system for modular automated NC integration) for the generation of bending sequences and assignment of tools for a die bending machine. The sheet metal parts are developed by a dedicated part modeler. In this modeler, the volume model of a sheet is converted into foil type of body. The material thickness which is retained as an attribute to the foil body, is used in the development of original volume model, so that the bending zones are converted from sharp bends present in the foil body into radius bends. The bend zone is represented as edges with attributes representing geometrical and technological data. The combination of part modeler and expert system technique is used to generate a feasible sequence for components that can be manufactured on a die bending machine only. The system also simulates the bending process.

Smith et al., (1992) have developed FCAPP/SM, a feature-based relational database system for semi-generative process planning of sheet metal parts. The feature based system

has been developed using clipper language. The relational database has two main divisions, one for part and the other for keeping the records for each feature present in the part. The parts are represented by features generated through individual operations, and the operations identified are holes, slots and bends. An optimized process plan is selected using graph theoretic optimization by arranging all the features of the part into a predetermined precedence hierarchy.

Tilley (1992) has developed a production control system for manufacturing sheet metal components. The sheet metal part is represented by a structured description. This description is translated into operations which are classified as *needed* and *alternative* types. The process planning system has been divided into three stages viz.,

- During the initial tool selection, all possible manufacturing methods will be considered
- Based on the list of methods, the scheduler selects a machine or a group of machines for producing the part
- Based on the machine characteristics (set up, turret size, etc.) and the set of available tools, the individual tools and manufacturing methods are selected in real time

Tools required for nibbling operation are selected manually. Details of the procedure followed for the selection of tools for other shearing operations are not given.

Vidlicka (1993) has developed a computer aided manufacturing system for finding the bending sequence, after developing the flat pattern of the $2\frac{1}{2}$–D part. The system takes the part information interactively from its left hand open sided end, on the basis of point to point, and the part should be of open type.

Palani et al., (1994) have developed a knowledge based simulation approach for sheet metal forming called intelligent design environment (IDE). In this, the part is defined by its geometry through the CAD interface. Neither the geometric modeler used for this purpose, nor the features identified from the part are given in the paper. Lin and Peing (1994) have investigated the application of expert system in sheet metal bending design. The bending features are input to the system interactively.

de Vin et al., (1994) have generated bending sequences for sheet metal components in a CAPP environment. This system is a part of the bigger system called PARTS-S, which is entitled *the automated process planning for small batch part manufacturing, based on form features and the technology of sheet metal processes.* The system considers punching, nibbling, laser welding, laser cutting and air bending. Information regarding the CAD system, and the type of CAD interface is not given, except stating that the features are imported from the external CAD system via a CAD interface.

Lentz and Sowerby (1994) have extracted holes from sheet metal components present in the convex, concave or in the intersection of these regions. The component is represented in the CAD database by B_rep modeler. The part is divided into convex, concave and transition regions (Lentz and Sowerby, 1993). Using modified face adjacency hypergraph (MAFH), two methods have been suggested. One for extracting convex holes, and the other for extracting general cylinder and general cone type of holes.

1.4 Scope and statement of purpose

Survey of the available literature reveals that the research in the area of feature recognition for sheet metal parts is not significant, even though they form a very significant part of the metal working processes, and are also widely used in almost all types of industries.

The current literature on feature recognition for sheet metal components indicates that most of the researchers have used solid modeling for feature recognition. They have converted the component created by the solid modeler into a foil type of component by neglecting its thickness, and then the pressworking features are identified. The combination of geometric information about the component and graph based approach provides a good platform to recognize features created by the solid modeler.

However, in practice, most of the industries use wireframe modeling to represent sheet metal components because wireframe modeling is easy to create and it can represent the sheet metal components unambiguously. In the present work, the ambiguity of wireframe modeler is overcome by representing the 2–D component or its layout (Fig. 1.24(a)) by the plan-view neglecting the thickness of the raw material, and in case of 3–D sheet metal component (Fig. 1.24(b)), the part is represented by the mean plane. Also, 3–D components considered are of open type (Fig. 1.24(b)). Using the world coordinate system or after converting the working coordinate system of entities into the world coordinate system, entity groups are identified using graph theoretic concepts. Additional information required for finding the location about entities and their groups are developed by using set theory and graph based approach. This helps in taking the decision about which entity groups are inside and which are outside with respect to one another. Also, entities present in each plane are identified. Thus the complete data structure of the physical entity i.e., component is developed from the simply itemized collection of vertices and edges in the CAD database. Other advantages of wireframe modeling are associated with the cost of hardware and software. Also, combination of geometric information about the component, along with set theory and graph theory are used to recognize pressworking features from the CAD database.

The literature review also reveals that in the feature recognition process, usually form

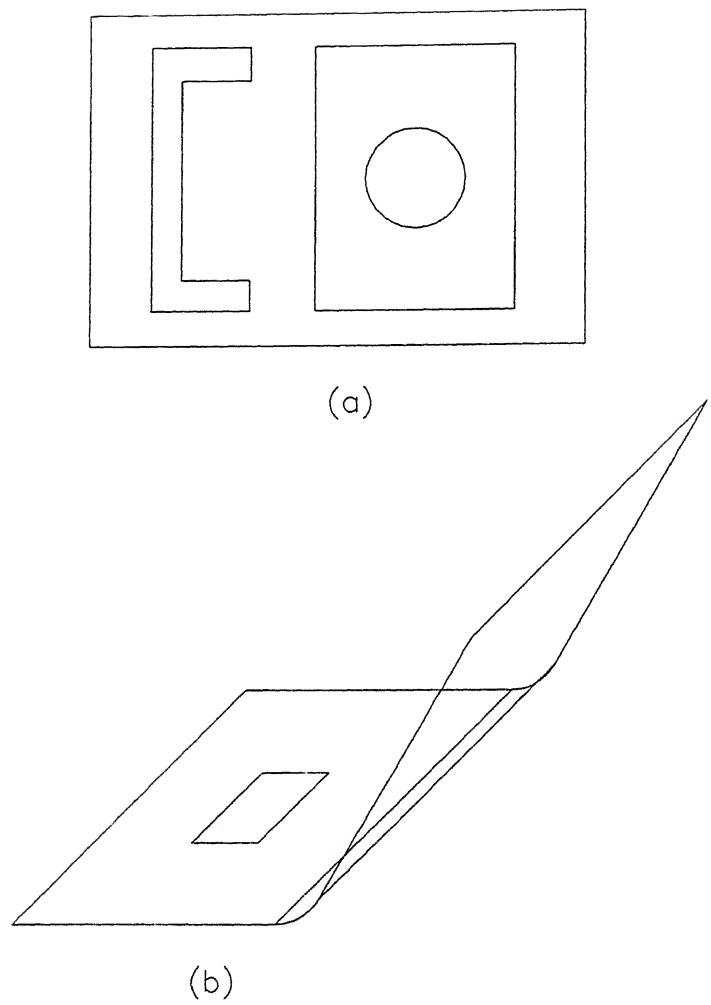


Figure 1.24: Wireframe model representation of sheet metal component and part layout in the present work: (a) 2–D part layout, and (b) 3–D component.

features are identified first from the design database, and then manufacturing features. *In case of sheet metal components, form features produced during pressworking operations depend on tool shape, size and force applied on it, and not on the relative movement between the tool and workpiece as in case of conventional metal cutting operations.* Therefore, in the present work, manufacturing features are identified directly from the design database. Further, pressworking operations like cutoff, notching, parting, shearing and slitting are also dependent on the parameters like, type of raw material (coil, sheet and strip), type of layout and direction of layout. These operations and parameters are not considered in most of the research work present in the literature on recognition of *pressworking features for sheet metal components.* These aspects are considered in this research work.

The *overall objective* of the present work is to recognize pressworking (manufacturing) features directly and automatically from the CAD database generated for a 3–D component and/or a 2–D component layout created by a wireframe modeler. The process of recognizing features involves the development of a set of generalized principles by reasoning and relating them geometrically and topologically using set theory and graph theory concepts.

The *specific objectives* of the present work can be stated as follows :

- To recognize pressworking features, by the proposed methodology, it is required to determine the *classification system of features* for developing a set of principles for characterizing them. Characterization of features helps in identifying them automatically from the CAD database in a hierarchical way. To classify pressworking features, the pressworking operations have to be classified first, as the features are directly related to the operations. Therefore, a classification system for operations is proposed, which is subsequently extended for classification of features.

- To identify shearing and forming features, along-with the geometrical parameters, directly from the CAD database of the component and/or part layout created flexibly in wireframe modeling. The features so identified will give alternative methods by means of which a component can be manufactured from the raw material.

- To develop the flat pattern for a formed component by taking thinning effect into consideration for recognizing shearing and forming features.

- To show the capabilities of the proposed methodology for identifying the shearing and forming features from the nested layout and the component respectively, with the help of illustrative examples, and real life components.

1.5 Organization of the dissertation

The dissertation is organized as follows:

Chapter 1 introduces the importance of feature recognition and discusses the various techniques of geometric modeling available. Literature review on feature recognition for rotational, prismatic and sheet metal components is given. At the end, scope and statement of purpose are given along with the organization of the dissertation.

Chapter 2 discusses the present and proposed classification systems for sheet metal components and pressworking operations. Definitions of various terminologies used in the present work are also given.

Chapter 3 discusses the methodology for the generation of a graph for a given component/part layout represented by the wireframe model and identification of subgraphs in the graph. A methodology for identifying the nature of the component (i.e., 2–D or 3–D) of the given drawing is also discussed.

Chapter 4 discusses the identification of feature sets, component set, and their elements.

Chapter 5 discusses the characterization of various shearing features, and their recognition process.

Chapter 6 discusses the characterization of various forming features, flat pattern development of a 3–D component, and identification of shearing features from the nested layout developed from the flat pattern.

Chapter 7 deals with the implementation of the above feature recognition methodology. It presents the outputs obtained for a few real life components and/or layouts.

In the last chapter, conclusions of the present work, and possible future extensions of this research work have been given.

Chapter 2

Classification Systems for Pressworking Features

2.1 Introduction

The term classification indicates grouping of objects with respect to certain criteria. The selection of criteria is based on the end use of the classification system. Objects present in a group or a family will have similar type of characteristic attributes. Classification of objects generally helps in the following areas (Gallagher and Knight, 1973) :

- Reduction of component numbers and feature varieties
- Simplification of process planning, estimating and costing
- Grouping of components for production
- Analysis of production procedures to remove anomalies
- Analysis of costing and estimating data to remove anomalies

Classification system also helps in tracing objects and identifying them from a group of objects or parts manufactured.

There are numerous types of classification systems that have been reported in the literature (Altan, 1983; Gallagher and Knight, 1973; Kang et al., 1993; Lascoe, 1988; Meinel, 1945; Nnaji et al., 1991; Groover, 1992). However, no individual system can be adopted as an universal one, as each system is developed with different criteria and for different purposes.

In mechanical engineering, the two major areas viz., design, and manufacturing (Groover, 1992), use different types of classification systems for codification of parts. The parts are classified into three categories based on part design and manufacturing attributes viz.,

1. Systems based on part design attributes,

2. Systems based on part manufacturing attributes, and

3. Systems based on both part design and manufacturing attributes.

The attributes of a part with respect to design and manufacturing typically included in classification systems are given in Table 2.1.

Part Design Attributes	
Basic external shape	Major dimensions
Basic internal shape	Minor dimensions
Length / Diameter ratio	Tolerances
Material type	Surface finish
Part function	
Part Manufacturing Attributes	
Major process	Operation sequences
Minor operations	Production time
Major dimensions	Batch size
Length / Diameter ratio	Annual production
Surface finish	Fixtures needed
Machine tools	Cutting tools

Table 2.1: Design and manufacturing attributes typically included in classification systems (Groover, 1992).

In this chapter, the classification systems for pressworking features for automatic recognition from CAD database have been discussed. The next section of this chapter gives existing general classification systems for sheet metal components and processes, and the proposed general classification system for pressworking operations. In the subsequent sections, definitions used in the present work for recognizing pressworking features are given, and the existing and proposed classification systems for shearing and forming operations and features are discussed.

2.2 General Classification Systems for Sheet Metal Components and Processes

Classification systems are categorized into two types viz., classification systems for sheet metal components, and classification systems for pressworking processes.

2.2.1 Classification systems for sheet metal components

The three major general classification systems (Gallagher and Knight, 1973) for sheet metal components used for codification purpose are :

1. Puschman classification system,
2. Opitz classification system, and
3. Salford classification system.

These three systems are based on overall and secondary shape feature criteria of components.

2.2.2 Classification system for pressworking processes

Pressworking processes are broadly classified into shearing and forming. Lascoe (1988) has classified sheet metal pressworking processes into four categories, viz., shearing, forming, drawing, and squeezing. The criterion used by Lascoe for classifying the processes is based on the working of metals. Table 2.2 gives the classification system for the same. The two types of classification systems for components and processes mentioned above are not suitable for automatic feature recognition from a CAD database, since the systems do not help in classifying a component either topologically or geometrically, and hence the design information cannot be translated into manufacturing information.

Nnaji et al., (1991) and Kang et al., (1993) have proposed a face oriented feature classification system (Fig. 2.1) for sheet metal fabrication of 3–D components created by B_rep solid model. The criteria used are :

1. Euler formula for features. is used for detecting whether a feature is a passage or
2. Topological configuration of the feature, and
3. Angle between the feature faces.

I Shearing operations

a. Blanking
b. Notching
c. Parting
d. Perforating
e. Piercing
f. Punching
g. Trimming

II Forming operations

a. Beading (Embossing)
b. Bending
c. Curling
d. Forming
e. Flanging
   1. Stretch flanges
   2. Shrink flanges
   3. Reverse flanges

III Drawing operations (Deep drawing)

IV Squeezing operations

a. Coining
b. Extrusion
c. Forging
d. Sizing
e. Swaging

Table 2.2: Classification system of sheet metal pressworking processes (Lascoe, 1988).

In the above classification system, the nested layout is not considered. Hence outside features i.e., operations performed outside the boundary of a component are not considered in case for shearing operations. Also in case of forming, features like deep drawing, spinning, and contour roll are not considered.

2.2.3 Proposed classification systems for pressworking processes and features

In view of the above shortcomings, a classification system (Fig. 2.2) has been proposed for pressworking processes to recognize features automatically. Based on this system, a classification system for entity groups is also suggested (Fig. 2.3). Information on entity groups is extracted from the CAD data base created by a wireframe model.

Pressworking operations are broadly classified into two categories viz., *Shearing and Forming.* These are classified based on the *topology of a form feature produced in the component by pressworking processes.*

Definitions and figures of various shearing and forming operations are given in Appendix A and Appendix B respectively.

Shearing operations are those, in which the work material is stressed beyond its elastic limit and fractured, when the ultimate strength is exceeded. Shearing features produced on a component and/or nested layout by pressworking operations are of two dimensional.

Forming operations are those, in which the work material is stressed above its elastic limit, but below its ultimate strength. The pressworking operations change the two dimensional work material into three dimensional part.

A general classification system for recognizing pressworking features from a CAD database of a component and/or layout is given in Fig. 2.3, where all the entity groups together portray a component and/or nested layout. Entity groups are categorized into two classes, viz., two dimensional and three dimensional. The entity groups represent subgraphs of a component graph and/or nested layout graph. The graph of a component and/or nested layout is of planar type.

Two dimensional entity groups represent the presence of shearing features, and every vertex of a subgraph is less than or equal to 3 degree. Whereas a three dimensional entity group represents the presence of forming features, and some of the vertices of the subgraph will be greater than or equal to 3 degree.

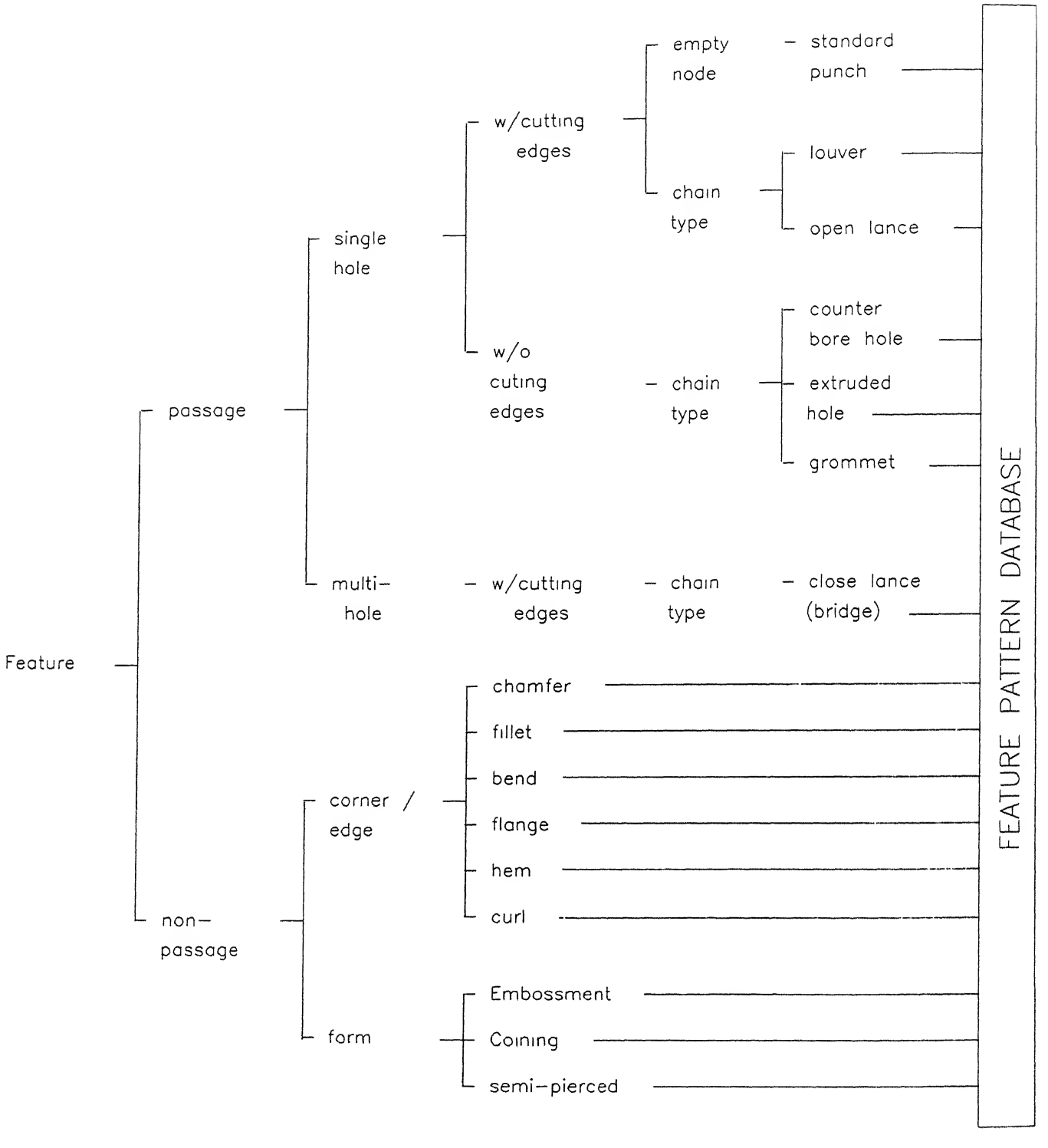


Figure 2.1: **Feature classification for sheet metal application (Nnaji et al., 1991).**

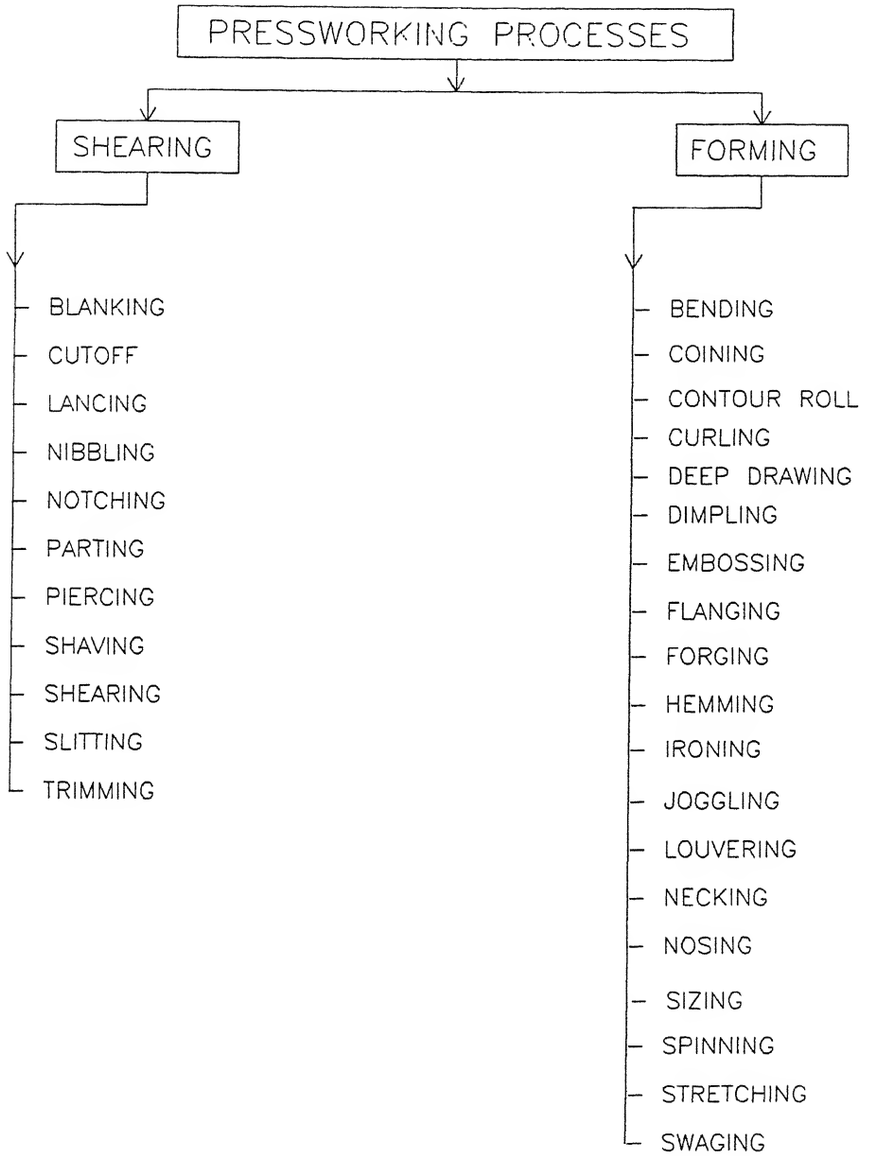


Figure 2.2: General classification system for pressworking processes.

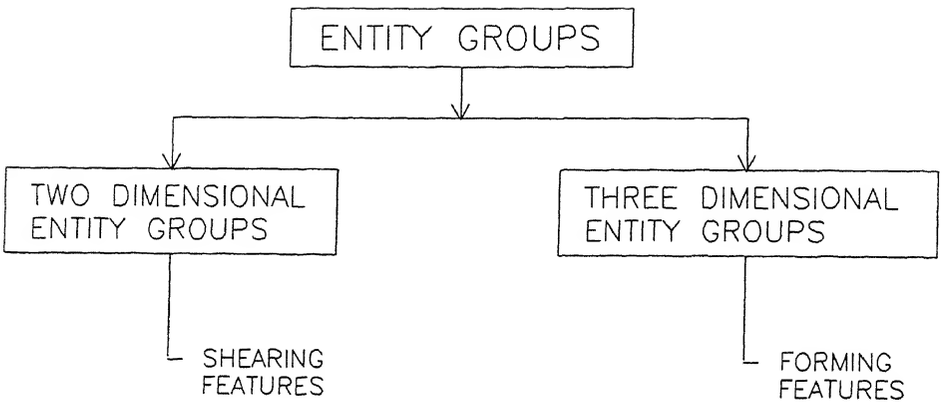


Figure 2.3: General classification system for pressworking features.

2.3 Definitions

Definitions related to sheet metal components and/or nested layout are given below. These definitions use the plan of a 2–D nested layout and the mean plane of a 3–D component (Fig. 2.4) created by the wireframe model. Here the thickness of the sheet metal is neglected. The definitions are explained below :

Entity : An entity is an arc (a), circle (c), line (l), or a polyline (p).

Vertex (v) : A vertex is a point, center of an arc or a circle, or a point of intersection of a set of edges.

Degree (d) : The degree of a vertex is the total count of all its incident edges (Deo, 1990). In Fig. 2.4(c), vertices $v_1, v_2, ..., v_9$ are of degree 2, $v_{10}, ..., v_{12}$ are of degree 3, and v_{13} is of degree 0.

Concave vertex ($-$) : A vertex is concave, if the following condition is satisfied. A straight line joining any two points (one on each adjoining line), arbitrarily close to the vertex has all the points outside the component. (Graves et al., 1989, source Sahay et al., 1990).

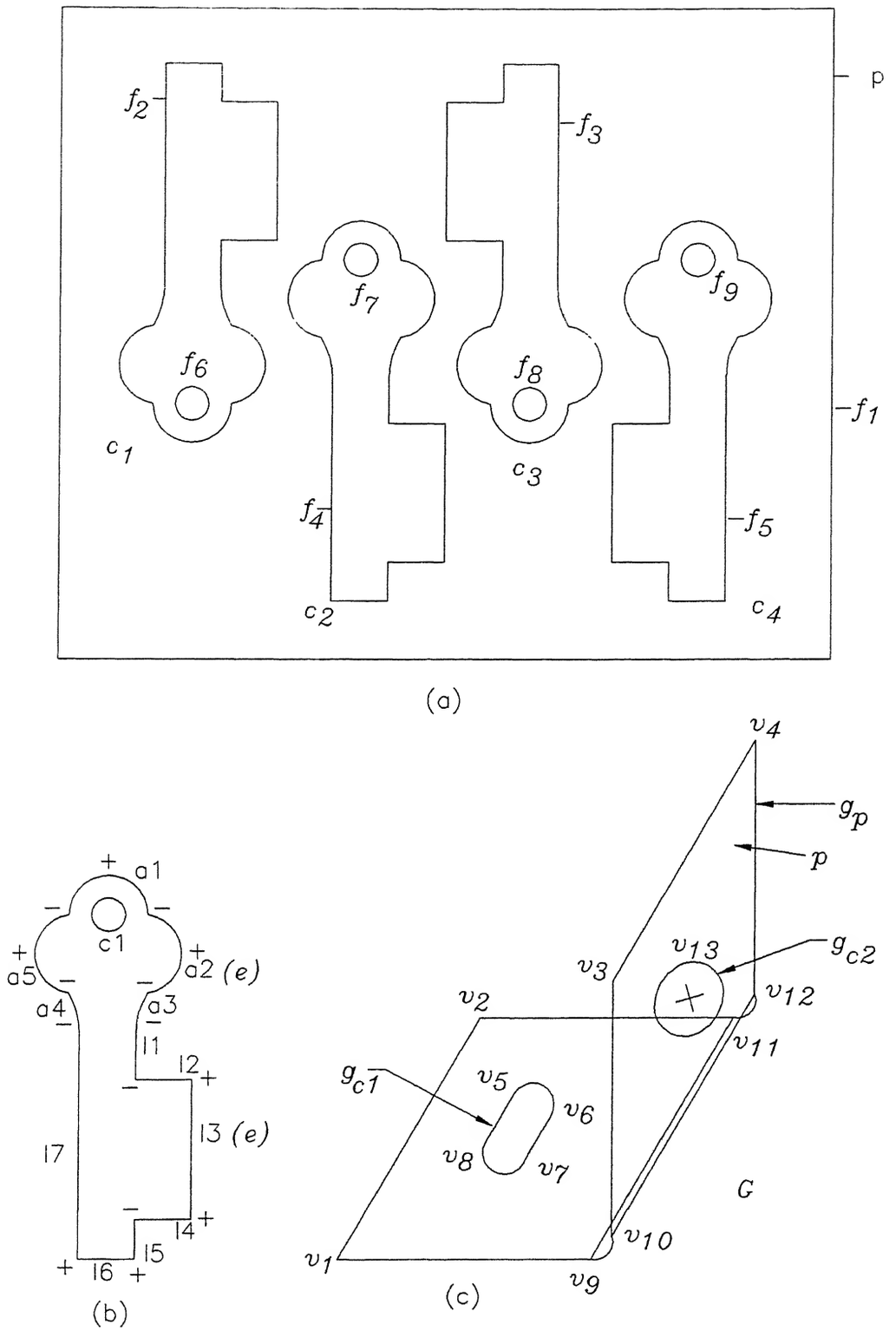


Figure 2.4: Representation of sheet metal components in a wireframe model : (a) Plan of a 2–D nested layout, (b) A 2–D component, and (c) Mean plane of a 3–D component.

Convex vertex (+) : Under similar conditions as in concave vertex, if all the intermediate points are inside the component, then the vertex is convex. (Graves et. al., 1989, source Sahay et al., 1990).

Edge (e) : An edge is an undirected line segment or arc, and connects two distinct end points (Deo, 1990).

Graph (G) : A graph is a set of edges and vertices of a component and/or nested layout, such that no two edges have a point in common which is not a vertex.

Cycle or Loop (L) : A cycle or loop is a closed sequence of edges in which the each edge appears once and every vertex is of degree two (Deo, 1990). In Fig. 2.4(b), $l1, l2, ..., a5, a4, a1, a2, a3$ forms a loop.

Face or plane (P) : A face is an ordered, oriented cycle of edges, such that all of the edges in a given cycle are shared with one and only one another cycle in a graph (Deo, 1990).

Parent subgraph (g_p) : A parent subgraph is a component of the graph. Geometrically all other components of the graph will be inside the parent subgraph.

Child subgraph (g_c) : Child subgraph is also a component of the graph, however, geometrically the child subgraph is inside the parent subgraph.

Feature (f): A feature represents a group of entities in an ordered form, which can be produced by sheet metal pressworking operations.

Feature set (F) : A feature set consists of features and forms a spatially disjoint subgraph of features in case of shearing operation feature. $F = \{f_1, f_2, ..., f_9\}$ in Fig. 2.4(a) represent features set.

Raw material feature set (R) : A raw material feature set consists of a feature, such that all other features of the feature set lie inside it. Geometrically, its projected length along X and Y directions are maximum among all features. Every vertex of the graph is less than 3 degree. $R = \{f_1\}$ in Fig. 2.4(a).

Boundary feature set (B) : A boundary feature set is made up of features (also called as *parent features*) other than the feature included in the raw material feature set. These features does not lie inside any feature other than the element of R. The vertices of the feature subgraph are less than or equal to 3 degree. $B = \{f_2, ..., f_5\}$ in Fig. 2.4(a).

Inside feature set (N) : An inside feature set consists of features (also called as *child features*) which are not included in raw material feature set and boundary feature set. Also, the verities of the feature subgraph are less than or equal to 3 degree. $N = \{f_6, ..., f_9\}$ in Fig. 2.4(a).

Component set (C) : A component set consists of components. The element of the component set is in turn a set consisting of a parent feature and child features located inside the parent feature. $C = \{c_1, ..., c_4\}$ in Fig. 2.4(a), where $c_1 = \{f_2, f_6\}; ..., c_5 = \{f_5, f_9\}$.

2.4 Classification system for shearing operations and features

The proposed classification system for shearing operations (Fig. 2.5) help in automatic feature recognition process from CAD database. Shearing operations are classified into three categories viz.,

1. Inside operations, which are performed within the boundary of a component,
2. Boundary operations, which are performed along the boundary of a component, and
3. Outside operations, which are performed outside the boundary of a component.

Inside and boundary operations are further categorized into rough and/or semifinish, and finish operations. Rough and/or semifinish operations are those operations for which conventional dies readily produce parts within a total tolerance of 0.05 mm to 0.25 mm, depending on the accuracy of the dies and the condition of the press (Metals Handbook, Vol.4, 1969). Finish operations are required for parts requiring a total tolerance lower than that of mentioned above (i.e., of higher accuracy).

In the proposed classification system, nibbling, shaving, and trimming operations are further categorized into blanking and piercing type viz., pierce nibbling, blank nibbling, pierce shaving, blank shaving, pierce trimming, and blank trimming operations. These categories are again based on the performance of operations in a component. For example, if nibbling operation is performed to get an obround hole in a component, the operation is called pierce nibbling. On the other hand if nibbling is performed to get the blank of the component, then it is called blank nibbling. Similarly other operations are also classified.

Based on the classification system for shearing operations, shearing features are classified. The proposed classification system for shearing features is given in Fig. 2.6. The

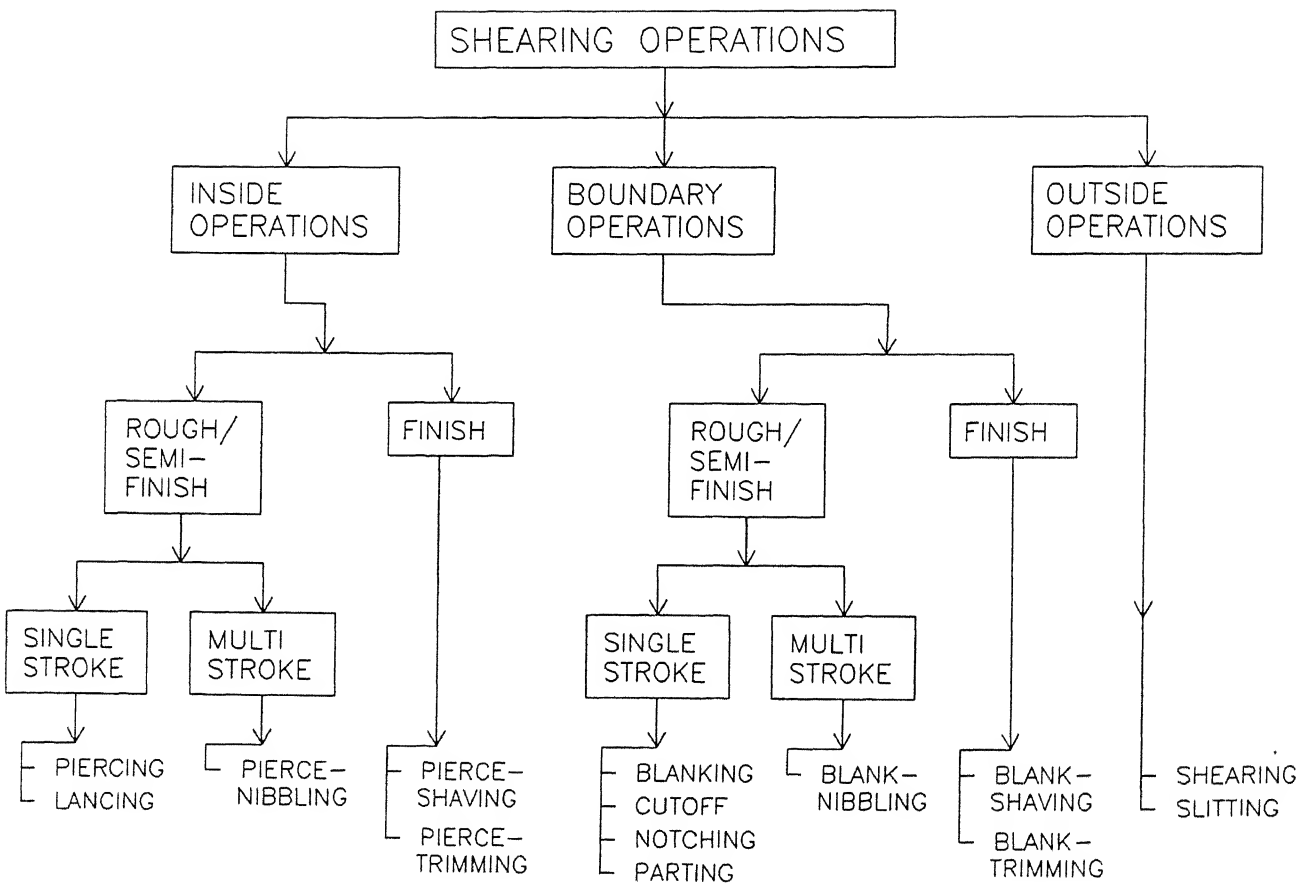


Figure 2.5: Classification system for shearing operations.

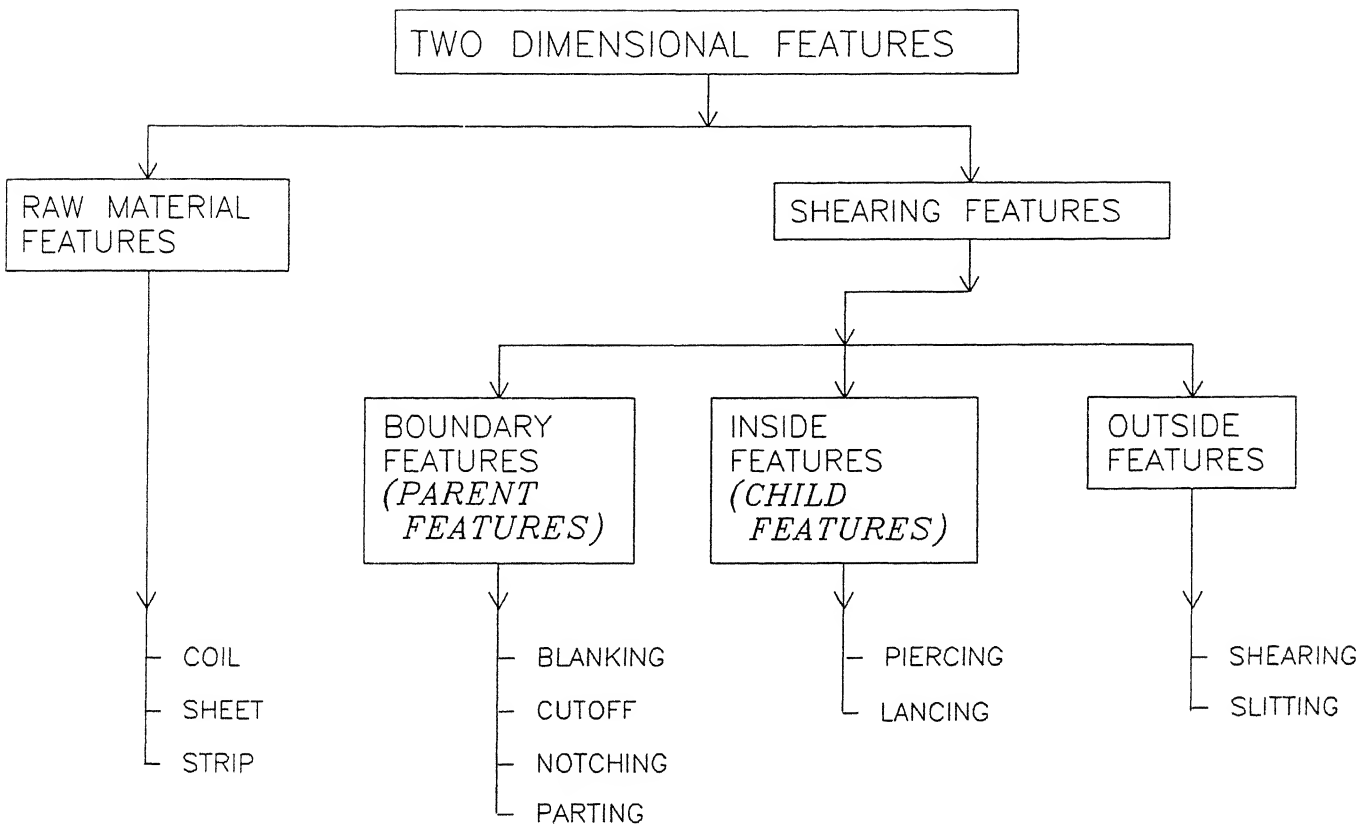


Figure 2.6: Classification system for shearing features.

two dimensional features are classified into two categories viz., raw material feature and shearing features.

Raw material feature represents the type of raw material used for a component or nested layout. The three types of raw material features considered are coil, sheet, and strip.

Shearing features represent group of entities in an ordered form, and these features are produced by shearing operations. Shearing features are further categorized into three categories viz.,

1. Inside features (or child features), which are present within the boundary of a component,

2. Boundary features (or parent features), which are present along the boundary of a component, and

3. Outside features, which are present outside the boundary of a component.

2.5 Classification system for forming operations and features

The classification systems available in the literature for forming operations are discussed below :

Meinel (1945) (source: Sachs, 1966) has classified formed sheet metal parts into five categories (Table 2.3), based on the contour of the finished part. The system is generally meant for use in aircraft industries. The parts that appear in the border line of two or more categories are classified on a quantitative basis, rather than qualitative basis.

Altan et al., (1983) have classified sheet metal forming processes into five categories. They have also used the contour of the finished parts as the criterion for classification, and their proposed classification system is given in Table 2.4.

The above classification systems are not useful in the environment of automatic manufacturing feature recognition from a CAD database, since the systems do not classify the finished parts or processes based on either topology or geometry of the sheet metal components. Hence it is not possible to automatically transfer the design information into manufacturing information.

In view of shortcomings of the existing classification systems, a new classification system is proposed for forming operations (Fig. 2.7). This system helps in converting automatically the design information from a CAD database into manufacturing information. The criteria used are :

I Singly curved parts
- a. Straight sections
- b. Straight flanged parts
- c. Single-curvature, smoothly contoured parts

II Contoured flanged parts
- a. Parts with stretch flanges
- b. Parts with shrink flanges
- c. Parts with reversely contoured flanges

III Curved sections
- a. Slender curved sections of uniform cross-sectional contour
- b. Compact curved sections
- c. Slender curved sections on nonuniform cross-sectional contour

IV Deep-recessed parts
- a. Deep-recessed parts with vertical walls
  1. Cup-shaped parts
  2. Tubular parts
  3. Box-shaped parts
- b. Deep-recessed parts with slopping walls
  1. Closed parts with sloping walls
  2. Open parts
  3. Semi-tubular parts

V Shallow-recessed parts
- a. Double-curvature, smoothly contoured parts
- b. Dish-shaped parts
- c. Beaded, embossed, and corrugated parts

Table 2.3: Sheet metal classification system based on the contour of finished parts (Sachs, 1966).

I Bending and Straight flanging
- a. Break bending
- b. Roll bending

II Surface contouring of sheet
- a. Contour stretch forming (Stretch forming)
- b. Andro forming
- c. Age forming
- d. Bulging
- e. Creep forming
- f. Die quench forming
- g. Vaccum forming

III Linear contouring
- a. Linear stretch forming (Stretch forming)
- b. Linear roll forming (Roll forming)

IV Deep recessing and flanging
- a. Spinning (and roller flanging)
- b. Deep drawing
- c. Rubber pad forming
- d. Marform process
- e. Rubber diaphragm hydroforming

V Shallow recessing
- a. Dimpling
- b. Drop hammer forming
- c. Electromagnetic forming
- d. Explosive forming
- e. Joggling

Table 2.4: Classification system for sheet metal forming processes, based on the contour of the finished parts (Altan et al., 1983).

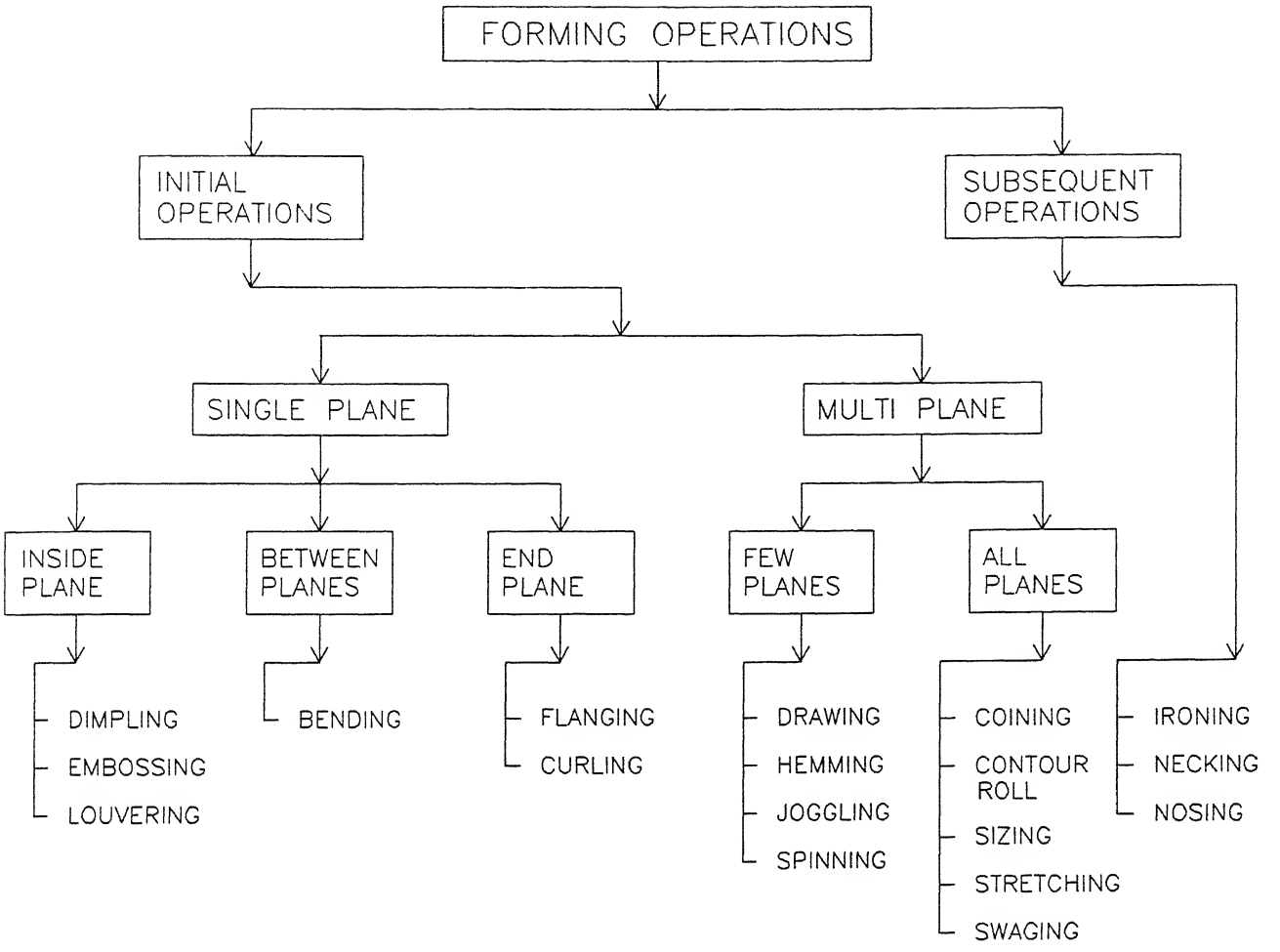


Figure 2.7: Classification system for forming operations.

1. Preceding operation relationships, and

2. Number of planes of a component subjected for forming operations.

Based on the preceding operation relationships criterion, forming operations are further categorized into initial, and subsequent operations.

Initial operations are those, which do not have any preceding operation and are performed on the work material to obtain near net shape of a component.

Subsequent operations are those, which are required for obtaining the final shape of the component and are performed only after initial operations have been carried out.

Initial operations are further classified into two divisions based on the number of planes of a component simultaneously used during a forming operation. They are :

1. Single plane operation, performed on only one plane of a component, and
2. Multi-plane operation, performed on more than one plane of a component simultaneously.

Operations of single plane type are categorized into three classes based on their location in a component. They are :

- Inside plane operation, performed within the plane of a component
- Between planes operation, performed between two planes of a component
- End plane(s) operation, performed at an end plane(s) of a component

Multi-plane operations are categorized into two divisions based on the number of planes used during the operation, viz.,

- Few-planes operation, performed on a few planes of a component simultaneously
- All-planes operation, performed on all planes of a component simultaneously

Based on the above proposed classification system for forming operations, classification system for recognizing forming features from 3–D entity groups is given in Fig. 2.8. Features present in 3–D entity groups are classified into three types viz., parent subgraph feature, child subgraph feature, parent-child subgraph feature. These features are present in their respective subgraphs of the component graph.

Parent features are further categorized into single plane features and multi-plane features. Single plane feature is present in only one face (or plane) of the parent subgraph of a component, where as multi-plane feature is present in more than one face (or plane) of the parent subgraph of a component

Single plane features are categorized into two classes. They are :

- Between planes feature, present between two planes of the parent subgraph of a component
- End planes feature, present at end planes of the parent subgraph of a component

Similarly multi-plane features are categorized into two divisions. They are :

- A feature present in a few planes of the parent subgraph of a component is called as *few planes feature.*
- A feature present in all the planes of the parent subgraph of a component is called as *all planes feature.*

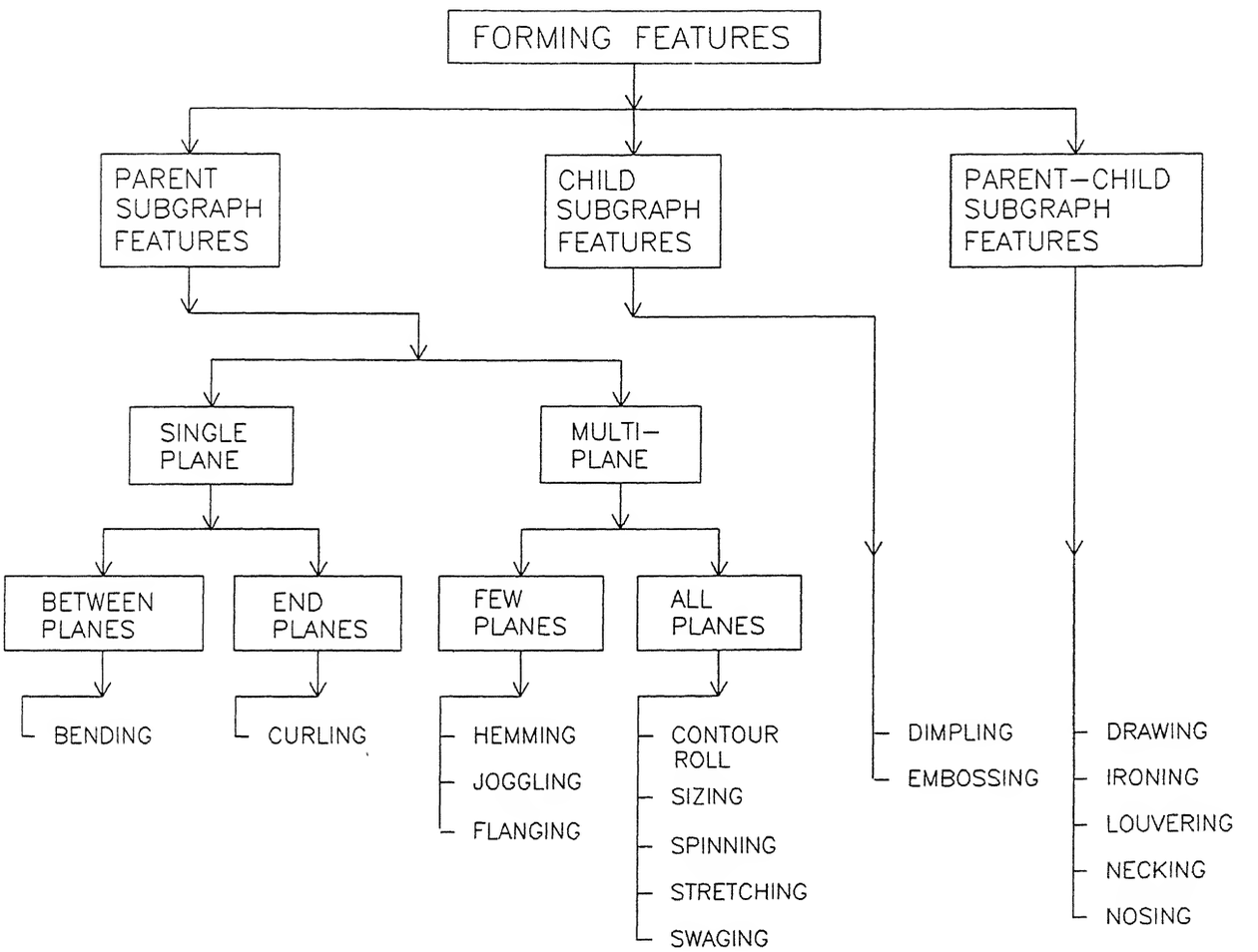


Figure 2.8: Classification system for forming features.

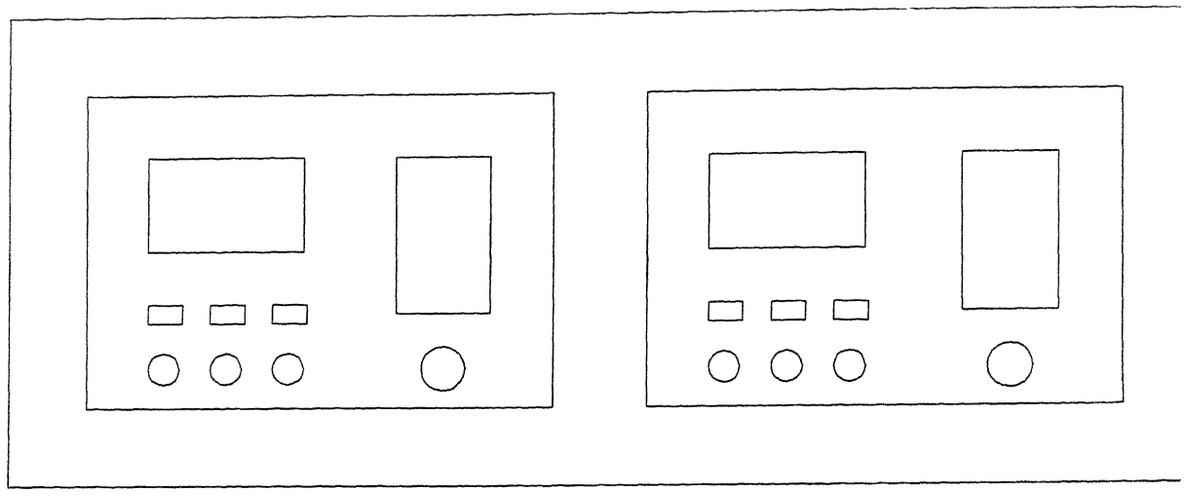
Chapter 3

Graph Generation and Subgraph Identification

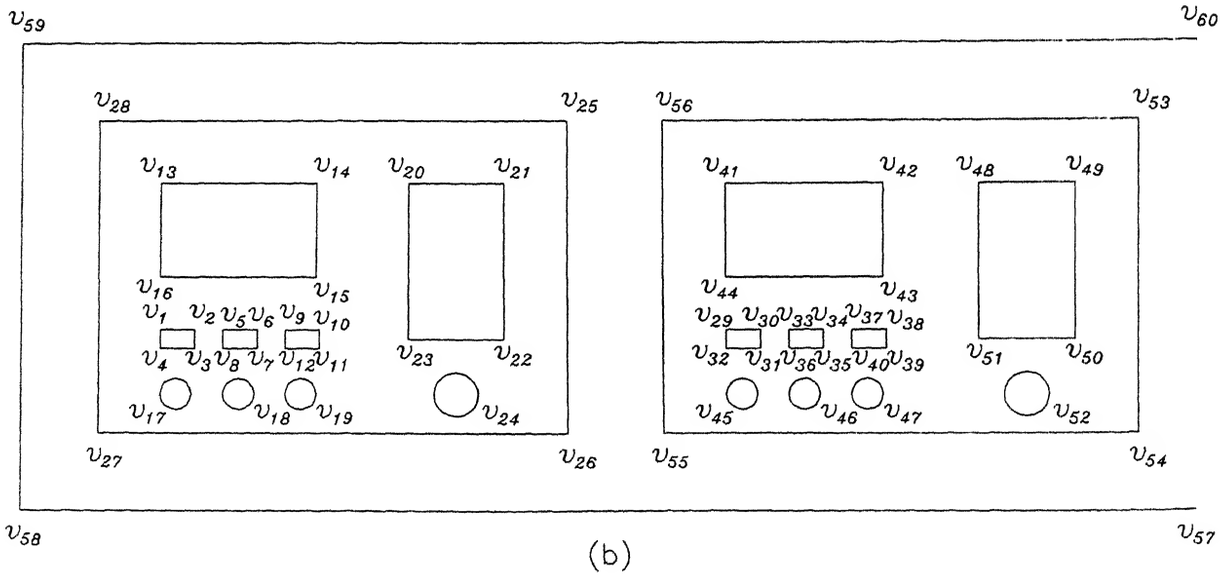
3.1 Introduction

A graph is defined as a set of edges and nodes with no two edges having a point in common which is not a vertex. Therefore, a graph is in essence a wireframe model minus the point coordinates and edge equations, i.e., the geometric information. In Fig. 3.1(a), a nested layout of a sheet metal component in 2–D, created using wireframe model is given and Fig. 3.1.(b) shows the graph of the same. Similarly, a sheet metal part created in 3–D using wireframe model and its graphical representation are given in Figs. 3.2.(a) and 3.2(b) respectively. From these figures, it can be seen that the wireframe and graphical representations are similar. Due to this similarity of topology between the part/component layout created in wireframe model and the graph of the same, the graph theoretic concepts have been used in the present work. In a wireframe model, the connectivity between entities is not maintained in the database. Also the flexibility of a designer/draftsman is not constrained in the creation of a component/part layout in the present work. Therefore, the graph of the component/part layout cannot be established directly from the database because the standard data interchange (DXF) output file of AutoCAD Release–10 is used as input to the system. To get connectivity between the entities, *repeated vertices* between entities have to be determined and should be merged into one vertex. Thus, the graph of the component/part layout obtained will satisfy Euler's formula for wireframe model representation. From the component/part layout graph, vertex adjacency (VA) matrix is obtained. Using VA matrix as input to the system, vertices present in the entity groups are determined by *vertex fusion methodology.* The entity groups represent the *subgraphs* of the component/part layout graph, and thus, the subgraphs of component/part layout graph are determined. With the help of degree of vertices present in the graph, the nature

CENTRAL LIBRARY
I. I. T., KANPUR
Acc. No. A121983



(a)



(b)

Figure 3.1: Nested layout of a 2–D component : (a) Plan view, and (b) Layout graph.

of a component/part layout is determined, i.e., whether the graph is established for a 2–D part layout, or for a 3–D component.

3.2 Definitions

The following definitions of graph theoretic concepts (Deo, 1990; Ganter and Skoglund, 1993; and Agarwal and Waggenspack Jr., 1992) are not meant to be exhaustive, but are given to provide a common vocabulary (a few are repeated from Chapter 2) to aid the clarity of discussion.

Edge : An edge is undirected line segment or arc, which connects two end points.

Node : A node is an end point for any given number of edges.

Degree : The degree of a vertex is the total count of all its incident edges. For example, the degree of vertex v_1, v_4, and v_{19} in Fig. 3.2(b) are 2, 3, and 0 respectively.

Path : A path is an alternating sequence of nodes and edges (node, edge, node, ..., node, edge, node) that initiates at one node and terminates at another, and in which no vertex or edge occurs more than once. All vertices will have degree two, except for the two terminal nodes, which have degree one. In Fig. 3.1(b), edges present between the vertices v_{57}, v_{58}, v_{59}, and v_{60} along with vertices represents a path.

Cycle : A cycle (loop or circuit) is defined as a closed path, in which the initial vertex is also the final vertex. Every intermediate vertex of the closed path will have a degree of two, and appear only once. Edges present between v_{25}, v_{26}, v_{27}, and v_{28}, along with the vertices represents a cycle in Fig. 3.1(b).

Connected graph : A graph is said to be connected, if there exists a path between any two vertices of the graph.

Disjoint graph : A disjoint graph (disconnected graph) is the one in which there exists no path between two graphs, and these components of the graph are called subgraphs.

Adjacent vertices : Two vertices that share an edge are said to be adjacent to one another. Vertices v_1 and v_2 in Fig. 3.2(b) are connected by an edge, and hence are adjacent to each other.

Topology : This is defined as a formal study of those properties of space that remain invariant under a class of geometric transformation. Thus it provides wide class of shapes, irrespective of their exact geometry.

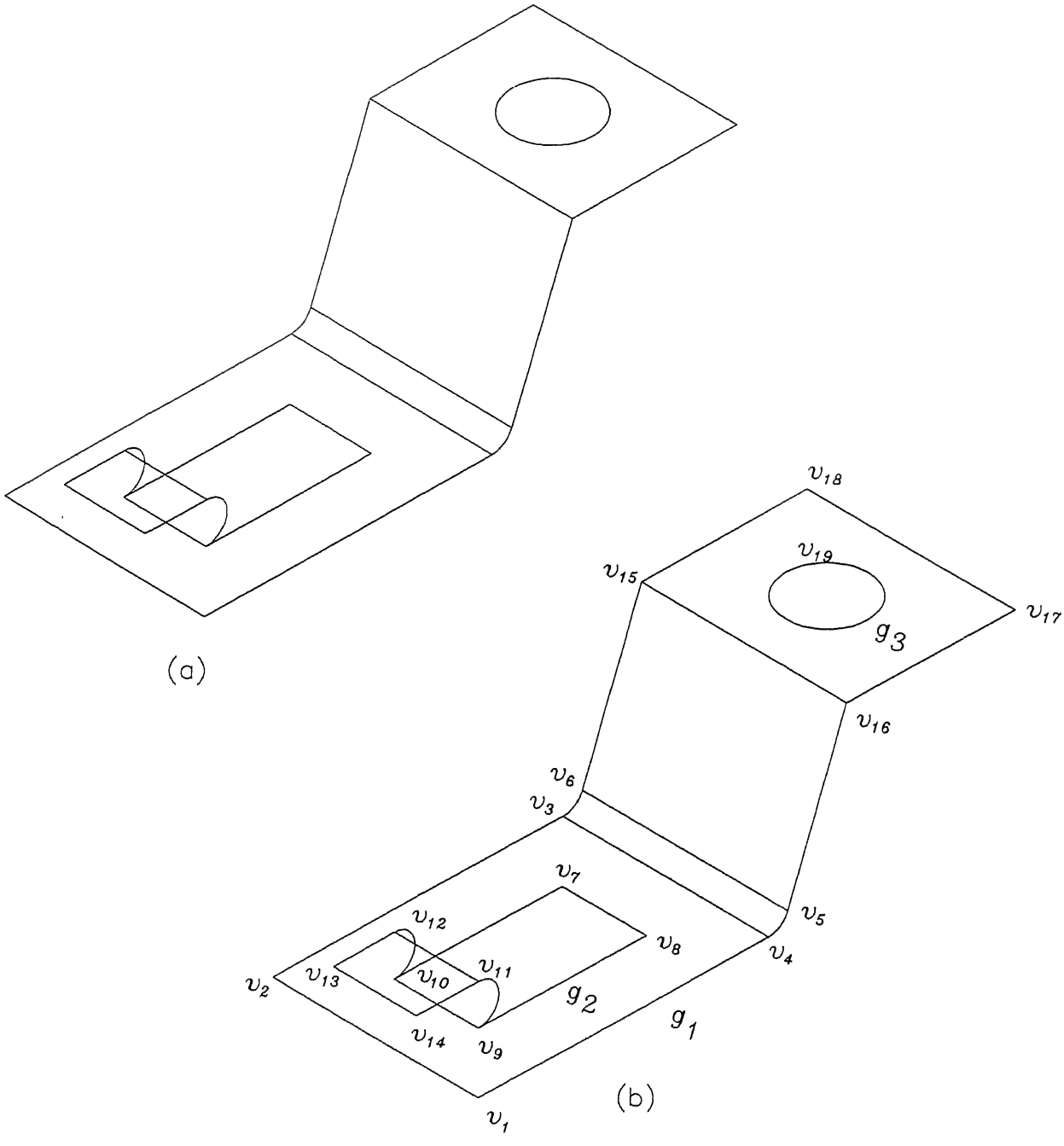


Figure 3.2: A 3–D sheet metal component : (a) Mean plane of the component, and (b) Component graph.

3.3 Types of entities used and assumptions made

Wireframe model is used in the creation of sheet metal part/component layout, and entities considered are given below :

1. An *arc* is associated with center coordinates, radius, start angle and end angle, i.e., v_c, rad, θ_s, and θ_e respectively as shown in Fig. 3.3.

2. A *circle* is represented by center coordinates (v_c) and radius (rad). In Fig. 3.3, v_c and rad represent the geometrical information available from the database for a circle.

3. A *line* is associated with start and end vertices. v_1 and v_2 represent the start and end vertices respectively in Fig. 3.3.

4. A *polyline* consists of series of vertices for lines (v_1, v_2, v_3, and v_4 in Fig. 3.3) and an arc segment is associated with start vertex (v_1), end vertex (v_2) and bulge (bl), where bulge is the tangent of one fourth of the included angle (θ) (i.e., $tan\,(\theta/4)$) of an arc segment as shown in Fig. 3.3.

While generating part/component layout, the following assumptions are made :

1. Drawing of a 2–D part for an optimum nested layout created in wireframe model representing the plan view is given as input to the system.

2. Drawing of a 3–D component created in wireframe model representing the mean plane of the component is given as input to the system.

3. Entities are drawn only once. The graph of the layout adheres to the *Euler's formula* for wireframe model (Wilson, 1985). This implies that

$$V - E + L - 2M = 0 \tag{3.1}$$

   where V, E, L and M represent the number of vertices, edges, loops and models respectively.

4. In coil type of raw material, the components drawn are just sufficient in number required for representing the optimum nested layout along length and the entity representing the end of the coil is not drawn.

5. Thickness of the component/layout is neglected while creating the wireframe model.

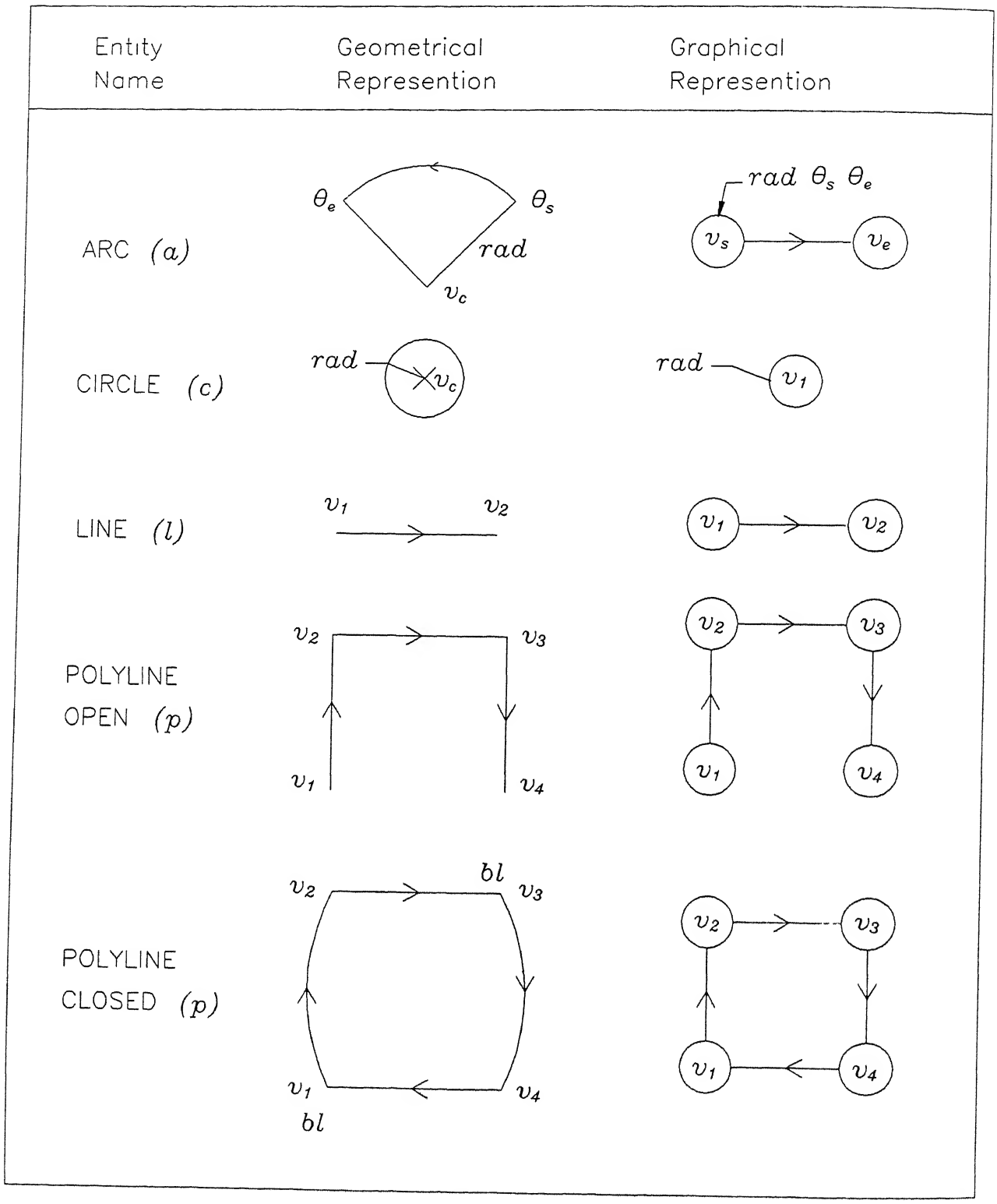


Figure 3.3: Entity names, and their geometrical and graphical representation.

3.4 Identification of duplicate vertices

Various entities considered for the present research is given in Fig. 3.3, along with their geometrical and graphical representation. From Fig. 3.3, it can be seen that if an entity group is made of polylines, then duplicate vertices are not present in the database. Since vertices are given in the order in which the entities have been created, the connectivity between edges or information about adjacent vertices is obtained directly from the database. In case of a circle, there is only one vertex, which represents the center coordinate, and hence duplicate vertices are not present. The circle entity represents by itself an entity group. In case of arcs and lines, duplicate vertices have to be identified, since individual entities have their own geometrical information. However, information on connectivity between entities is absent.

In case of an arc, the geometrical information available are center coordinates, radius, start angle and end angle. Information about start vertex and end vertex is not available. Therefore, for every entity arc, its start and end vertices are determined as given below :

Let v_c and rad denote the center coordinates and radius of an arc respectively obtained from the database. Let v_s and v_e represent start and end vertices respectively. The x and y coordinates of the start vertex (v_s^x and v_s^y, respectively) are given by

$$v_s^x = v_c^x \pm rad\, cos(\theta_s) \qquad (3.2(a))$$

$$v_s^y = v_c^y \pm rad\, sin(\theta_s) \qquad (3.2(b))$$

where v_c^x and v_c^y are x and y coordinates of the center of an arc and θ_s is its start angle. Similarly, x and y coordinates of end vertex (v_e^x and v_e^y) are given by

$$v_e^x = v_c^x \pm rad\, cos(\theta_e) \qquad (3.3(a))$$

$$v_e^y = v_c^y \pm rad\, sin(\theta_e) \qquad (3.3(b))$$

where θ_e is the end angle of the arc under consideration.

In the above equations, θ_s and θ_e are measured in the counterclockwise direction, with East or X axis direction as a reference direction (Fig. 3.4). Also, positive or negative sign in eqns. (3.2) and (3.3) is dependent on the quadrant in which the arc starts and/or ends. In eqns. (3.2(a)) and (3.2(b)), positive sign is considered if θ_s lies in I or IV quadrant, and negative sign if θ_s lies in II or III quadrant. This rule is also true for θ_e in eqn .3.3. Every arc and line entities have start vertex and end vertex. Therefore, in the database few vertices may be same. Hence, duplicate vertices have to be determined. Two vertices are said to be same, when their three coordinates (x, y, and z) are equal, i.e.,

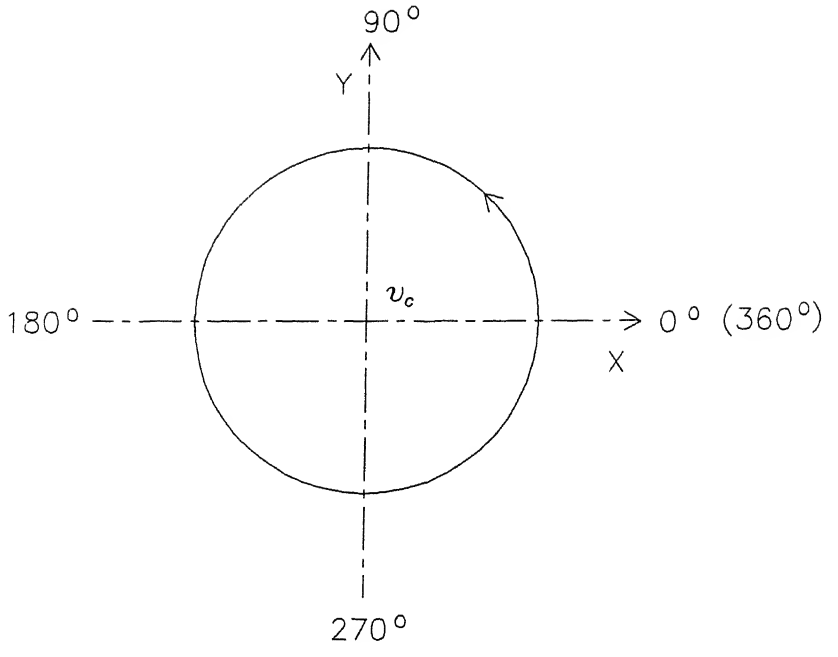


Figure 3.4: Arc measurement direction.

$$v_i = v_{i'} \quad \exists \quad \begin{aligned} v_i^x &= v_{i'}^x \\ v_i^y &= v_{i'}^y \\ v_i^z &= v_{i'}^z \\ i &\neq i' \end{aligned} \tag{3.4}$$

where v_i and v_i' are vertices belonging to the vertex set V_d having duplicate vertices. The superscript x, y, and z of v_i and $v_{i'}$ represent the x, y, and z coordinates of the vertices, respectively.

For example, for the sheet metal component shown in Fig. 3.5(a), the edges are made of lines and arcs. The edge set (E_d) of the component before duplicate vertices are determined is given by

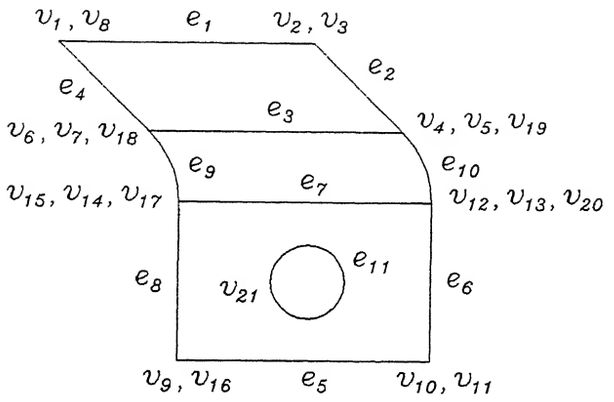
$E_d = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}, e_{11}\}$

and the element of edge set are duplex, having start vertex and end vertex as their elements, i.e.,

$e_1 = \{v_1, v_2\}$, $e_2 = \{v_3, v_4\}$, $e_3 = \{v_5, v_6\}$, ..., $e_9 = \{v_{17}, v_{18}\}$, $e_{10} = \{v_{19}, v_{20}\}$, and $e_{11} = \{v_{21}\}$. The vertex set V_d for the edge set E_d is given by

$V_d = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, ..., v_{21}\}$.

After duplicate vertices are determined and deleted from the set V_d, the remaining vertex set is called as $V_{d'}$. For the component graph shown in Fig. 3.5(a), duplicate vertices are removed and again labeled. The relabeled component graph is given in Fig. 3.5(b),



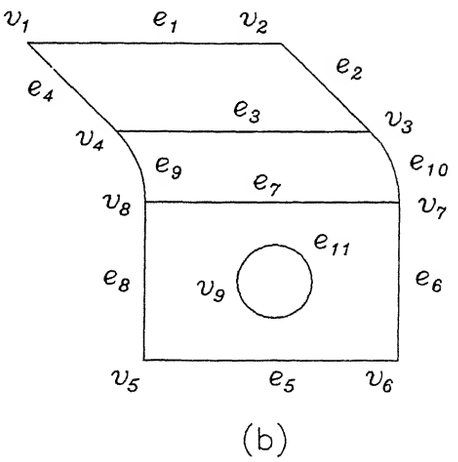


Figure 3.5: Component graph representing vertices and edges : (a) Graph before duplicate vertices are identified, and (b) Graph after duplicate vertices are identified, deleted and relabeled.

and the set $V_{d'}$ is given below :

$V_{d'} = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9\}$.

The edge set $E_{d'}$ of the component after duplicate vertices are identified, and relabeled is

$E_{d'} = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}, e_{11}\}$, and the duplex elements of edge set $(E_{d'})$ are given by

$e_1 = \{v_1, v_2\}$, $e_2 = \{v_2, v_3\}$, $e_3 = \{v_3, v_4\}$, ..., $e_9 = \{v_8, v_4\}$, $e_{10} = \{v_3, v_7\}$, and

$e_{11} = \{v_9, v_9\}$.

Thus, repeated vertices in a database are determined, and the component graph given in Fig. 3.5(b) adheres the Euler's formula given in eqn. 3.1, i.e.,

$V - E + L - 2M = 9 - 11 + 6 - 4 = 0$

where number of loops $(L)$is determined according to the Maze algorithm (Wilson, 1985).

3.5 Labelling of nodes and edges in the lexicographical order

The vertices identified and labelled Section 3.4 are randomly ordered. Therefore, the vertex set $V_{d'}$ of the component is also randomly ordered. This is due to wireframe model representation of the component/part layout, and flexibly created entity groups as well as entities within an entity group. The nodes of the graph (i.e., vertices) are labelled in lexicographical order, so that the direction and position of the components in the nested layout can be identified. The edges of the graph are also labelled in the lexicographical order which helps in the sequencing of edges in a graph, and for identifying pressworking features.

Vertices v_i and $v_{i'}$ are said to be in lexicographical order, if v_i precedes $v_{i'}$, i.e.,

$$\begin{aligned} v_i \prec v_{i'} \quad & \text{if } v_i^z < v_{i'}^z \text{ or} \\ & \text{if } v_i^z = v_{i'}^z, \text{ and } v_i^y < v_{i'}^y \text{ or} \\ & \text{if } v_i^z = v_{i'}^z, \; v_i^y = v_{i'}^y, \text{ and } v_i^x < v_{i'}^x. \end{aligned} \tag{3.5}$$

Similarly the duplex edges e_m and $e_{m'}$ are said to be in lexicographical order, if e_m precedes $e_{m'}$, i.e.,

$$e_m \prec e_{m'} \qquad \text{if } v_{s,m} < v_{s,m'} \text{ or } \quad \text{if } v_{s,m} = v_{s,m'}, \text{ and } v_{e,m} < v_{e,m'} \tag{3.6}$$

where start and end vertices (i.e., $v_{s,m}$ and $v_{e,m}$, respectively) are ordered in the duplex edge set e_m, such that $v_{s,m}$ precedes $v_{e,m}$ of the m^{th} edge.

3.5.1 Lexicographical ordering of vertices

In lexicographical ordering, vertices are initially sorted based on z coordinate values in ascending order. Then vertices of equal values of z coordinate are further sorted based on y coordinate values in ascending order. Again, for vertices having equal values of z and y coordinates, the vertices are sorted based on x coordinate values in ascending order, and the nodes of graph are labelled accordingly.

For example, the coordinate values of the elements of randomly ordered vertex set $V_{d'}$ of Fig. 3.5(b) are given in Fig. 3.6(a). The elements of $V_{d'}$ are lexicographically ordered (in accordance to eqn. (3.5)), and are given in Fig. 3.6(b). The vertex set V_{lo} given in Fig. 3.6(b) represents the lexicographically ordered vertex set. $(v_5, v_6, v_9, v_8, v_7, v_4, v_3, v_1, v_2)$. The vertices (elements) of the set V_{lo} are again labeled such that the indices of the elements are in the ascending order $(v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9)$, and this vertex set is denoted by V as given in Fig. 3.6(c).

3.5.2 Lexicographical ordering of edges

In lexicographical ordering of edges, the vertices of edges are again labeled in accordance with the the vertex set V. These vertices in an edge are rearranged such that the first vertex of an edge precedes the second vertex in the vertex set V. The first vertex is called start vertex $(v_{s,m})$ and second vertex is called end vertex $(v_{e,m})$ in a m^{th} edge, and this rearranged edge set is denoted by E_{ar}. The edge set E_{ar} is then lexicographically ordered in accordance to the eqn. (3.6) and the lexicographically ordered edge set is denoted by E_{lo}. The edges (elements) of E_{lo} is again labeled such that the indices of the elements are in ascending order, and this edge set is denoted by E.

For example, the elements of randomly ordered edge set $E_{d'}$ of Fig. 3.5(b) is given in Table 3.1. The elements (vertices) of edges belonging to the edge set $E_{d'}$ are rearranged, and the edge set E_{ar} is obtained (Table 3.2). This edge set E_{ar} is lexicographically ordered (in accordance to eqn. (3.6)), and the elements of the edge set E_{lo}

$(e_5, e_8, e_6, e_{11}, e_7, e_{10}, e_3, e_4, e_2, e_1)$

determined is given in Fig. 3.7(a). The edge set E_{lo} is again labeled such that the indices of the elements are in the ascending order

$(e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}, e_{11})$,

and this edge set is denoted by E as given in Fig. 3.7(b).

$V_{d'} = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9\}$

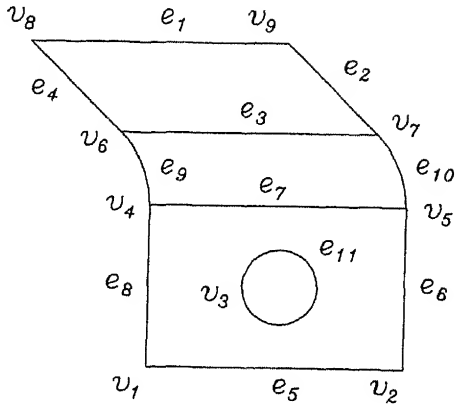
	x	y	z
v_1	100 00	25 00	75 00
v_2	135 00	25 00	75 00
v_3	135 00	5 00	75 00
v_4	100 00	5 00	75 00
v_5	100 00	0 00	50 00
v_6	135 00	0 00	50 00
v_7	135 00	0 00	70 00
v_8	100 00	0 00	70 00
v_9	117 50	0 00	60 00

(a)

$V_{lo} = \{v_5, v_6, v_9, v_8, v_7, v_4, v_3, v_1, v_2\}$

	x	y	z
v_5	100 00	0 00	50 00
v_6	135 00	0 00	50 00
v_9	117 50	0 00	60 00
v_8	100 00	0 00	70 00
v_7	135 00	0 00	70 00
v_4	100 00	5 00	75 00
v_3	135 00	5 00	75 00
v_1	100 00	25 00	75 00
v_2	135 00	25 00	75 00

(b)



$V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9\}$

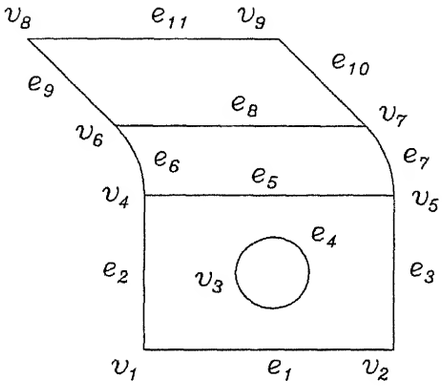
(c)

Figure 3 6· (a) Randomly ordered vertex set and its coordinate values, (b) Lexicographically ordered vertex set, and (c) Component graph and vertex set after labeling of lexicographically ordered vertex set again.

$$e_5 = \{v_1, v_2\} \qquad e_7 = \{v_4, v_5\} \qquad e_4 = \{v_6, v_8\}$$
$$e_8 = \{v_1, v_4\} \qquad e_9 = \{v_4, v_6\} \qquad e_2 = \{v_7, v_9\}$$
$$e_6 = \{v_2, v_5\} \qquad e_{10} = \{v_5, v_7\} \qquad e_1 = \{v_8, v_9\}$$
$$e_{11} = \{v_3, v_3\} \qquad e_3 = \{v_6, v_7\}$$

$$E_{lo} = \{e_5, e_8, e_6, e_{11}, e_7, e_9, e_{10}, e_3, e_4, e_2, e_1\}$$

(a)



$$e_1 = \{v_1, v_2\} \qquad e_5 = \{v_4, v_5\} \qquad e_9 = \{v_6, v_8\}$$
$$e_2 = \{v_1, v_4\} \qquad e_6 = \{v_4, v_6\} \qquad e_{10} = \{v_7, v_9\}$$
$$e_3 = \{v_2, v_5\} \qquad e_7 = \{v_5, v_7\} \qquad e_{11} = \{v_8, v_9\}$$
$$e_4 = \{v_3, v_3\} \qquad e_8 = \{v_6, v_7\}$$

$$E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}, e_{11}\}$$

(b)

Figure 3.7· (a) Lexicographically ordered edge set, and (b) Component graph and edge set after relabeling the above edge set

$$e_1 = \{v_8, v_9\} \qquad e_5 = \{v_1, v_2\} \qquad e_9 = \{v_4, v_6\}$$
$$e_2 = \{v_9, v_7\} \qquad e_6 = \{v_2, v_5\} \qquad e_{10} = \{v_7, v_5\}$$
$$e_3 = \{v_7, v_6\} \qquad e_7 = \{v_5, v_4\} \qquad e_{11} = \{v_3, v_3\}$$
$$e_4 = \{v_6, v_8\} \qquad e_8 = \{v_4, v_1\}$$

$$E_{d'} = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}, e_{11}\}$$

Table 3 1. First and second vertices of the edges are labeled according to the lexicographically ordered vertex set

$$e_1 = \{v_8, v_9\} \qquad e_5 = \{v_1, v_2\} \qquad e_9 = \{v_4, v_6\}$$
$$e_2 = \{v_7, v_9\} \qquad e_6 = \{v_2, v_5\} \qquad e_{10} = \{v_5, v_7\}$$
$$e_3 = \{v_6, v_7\} \qquad e_7 = \{v_4, v_5\} \qquad e_{11} = \{v_3, v_3\}$$
$$e_4 = \{v_6, v_8\} \qquad e_8 = \{v_1, v_4\}$$

$$E_{ar} = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}, e_{11}\}$$

Table 3.2· First and second vertices of the edges are rearranged

Thus, the lexicographically ordered vertices and edges of the component graph of Fig 3 5(b) are shown in Fig 3 7(b).

3.6 Connectedness and subgraph identification

Vertices present in a component/part layout graph, satisfying the Euler's formula are obtained from Section 3 4, but vertices present in each entity group (i e connectivity), and entity groups (i.e subgraphs) are not available directly from the database or from the lexicographical ordered vertices and edges Therefore, by fusion methodology (Deo 1990), connectedness and subgraphs in the component/part layout graph are determined. A pair of vertices are said to be connected, when an edge is present between them, i.e , they should be adjacent vertices. Subgraphs are disjoint graphs, i e , there is no existence of path between two graphs of the component/part layout graph

In the fusion process (Deo, 1990), a pair of vertices v_3 and v_4 in a graph are fused (merged) and the two vertices are replaced by a single new vertex v_{34}, such that every edge that was incident on either of vertices v_3 or v_4 or on both is incident on the new vertex Thus fusion of two vertices does not alter the number of edges, but it reduces the number of vertices by one (Fig 3 8)

In the following section, fusion methodology for determining connectedness and subgraphs of a component/part layout is given

3.6.1 Fusion methodology

In fusion methodology, initially, vertex adjacency (VA) matrix is obtained for the graph and any vertex can be considered as a starting point for fusion process All the vertices adjacent to the vertex under consideration are fused. The fused vertex is considered for further processing and all the vertices adjacent to it are fused The process of fusion is repeated until no more vertices can be fused. This indicates that a connected component has been fused to a single vertex If this exhausts every vertex of a graph, the graph is said to be connected and it does not have any subgraph. Otherwise, fusion process is continued with another new vertex (in a different subgraph) Thus, vertices present in an entity group, and different entity groups present in a component/part layout are identified and the flowchart for the same is shown in Fig. 3.9

In this fusion methodology, the maximum number of fusions that may have to be performed is $(n_v - 1)$, where n_v is a number of vertices present in a graph. In each fusion, utmost n_v logical additions may be performed Therefore, the upper bound on the execution time is proportional to $(n_v\ (n_v - 1))$.

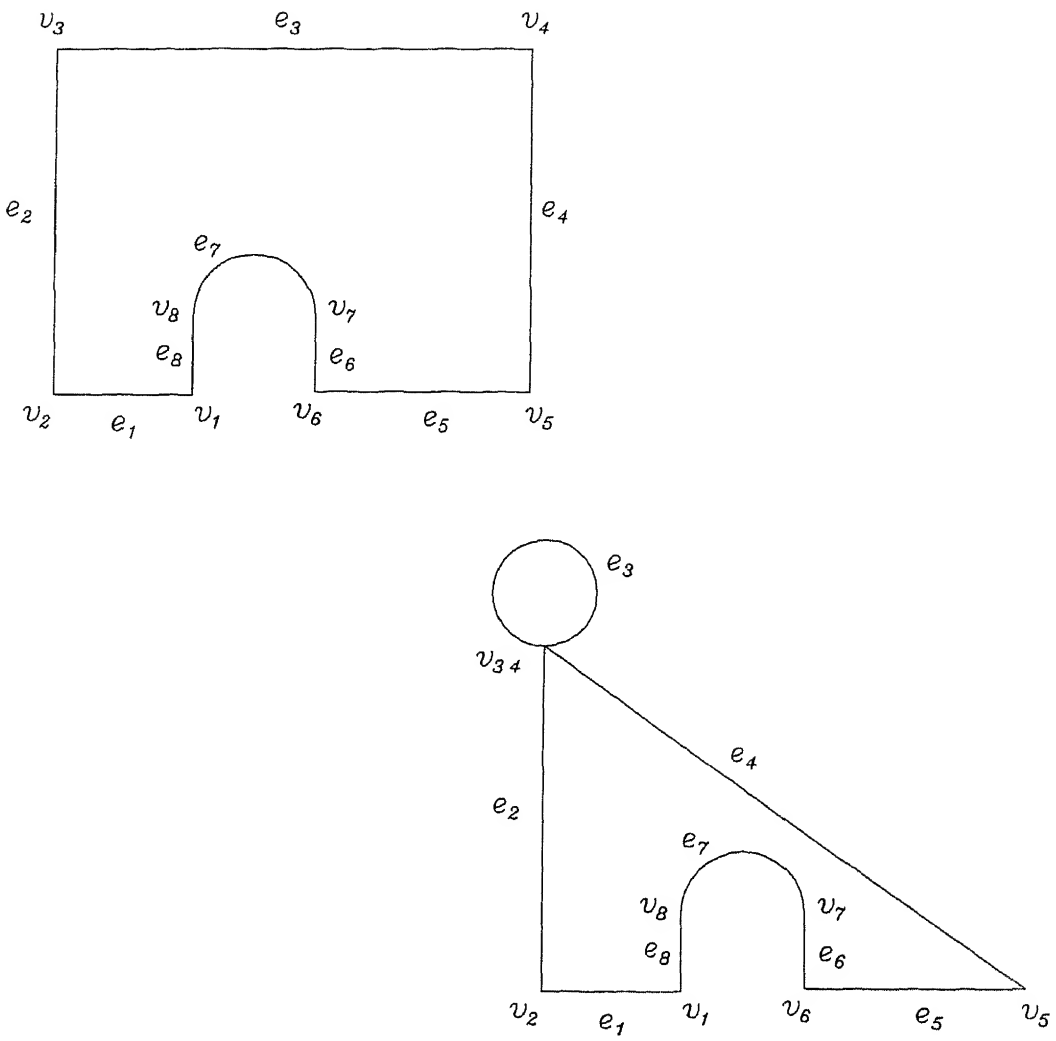


Figure 3 8: Fusion of vertices v_3 and v_4

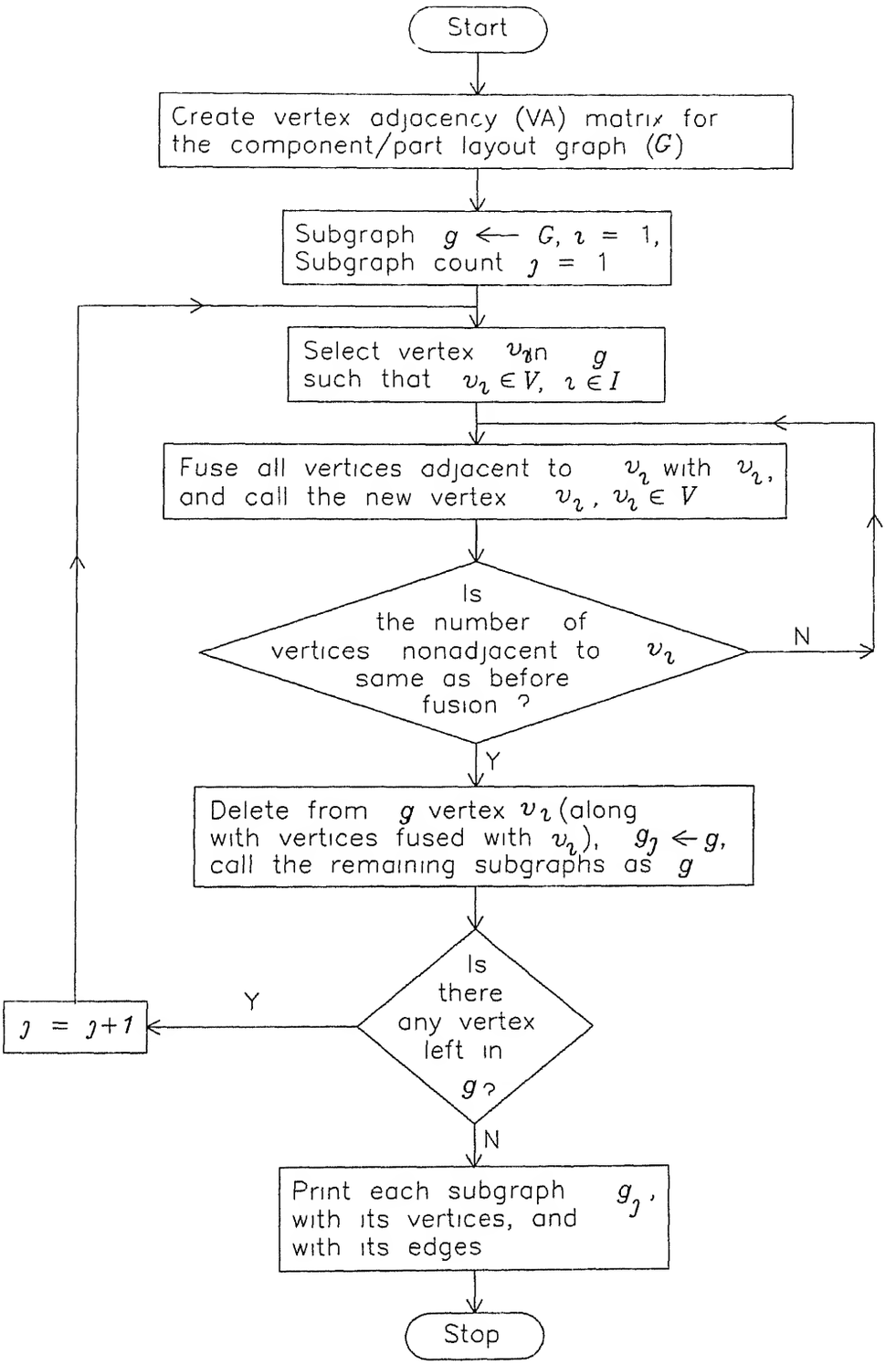


Figure 3.9· Flowchart for identifying connectedness and subgraphs in a component/part layout graph

For a component/part layout graph, a vertex adjacency matrix is obtained by analyzing edges A vertex is said to be adjacent to another vertex if an edge is present between them

Let the lexicographically ordered vertex set V of the component graph be made of vertices v_i, where i belongs to I, i.e.,

$$V = \{v_i \mid i \in I\} \tag{3.7}$$

where I is a set of subscripts of vertices (elements) of a vertex set V. An element of the set V is a triplet having x, y, and z coordinates, and is represented as :

$$v_i = \{v_i^x, v_i^y, v_i^z\} \tag{3 8}$$

The vertices are connected by edges. Let a duplex edge e_m be the m^{th} element of lexicographically ordered edge set E, having start vertex $(v_{s,m})$ and end vertex $(v_{e,m})$ as its elements, i e ,

$$E = \{e_m \mid m \in M\}, \tag{3 9}$$

and

$$\begin{aligned} e_m &= \{v_{s,m}, v_{e,m}\} \quad \text{or} \\ e_m &= \{v_{i,m}, v_{i',m}\} \quad i \in I, i' \in I, \text{ and } i' \neq i, \quad \forall\ m \in M. \end{aligned} \tag{3 10}$$

In eqns. (3.9) and (3.10), M is the set of subscripts of edge set E and $v_{s,m}$ precedes $v_{e,m}$ In eqn. (3.10), $v_{i,m}$ and $v_{i',m}$ represent $v_{s,m}$, and $v_{e,m}$ respectively.

Thus, two vertices are said to be adjacent, when vertices $v_{i,m}$ and $v_{i',m}$ are present in an edge e_m

From the edge set E, a VA matrix is obtained. For the Fig. 3 2(a), the component graph after the lexicographical ordering of vertices and edges is shown in Fig. 3.10(a), and the VA matrix is given in Fig 3.10(b). In the VA matrix, if the vertices are adjacent to each other, then the value of the cell is denoted by '1', else by '0' (represented by an empty cell in the VA matrix), i.e.,

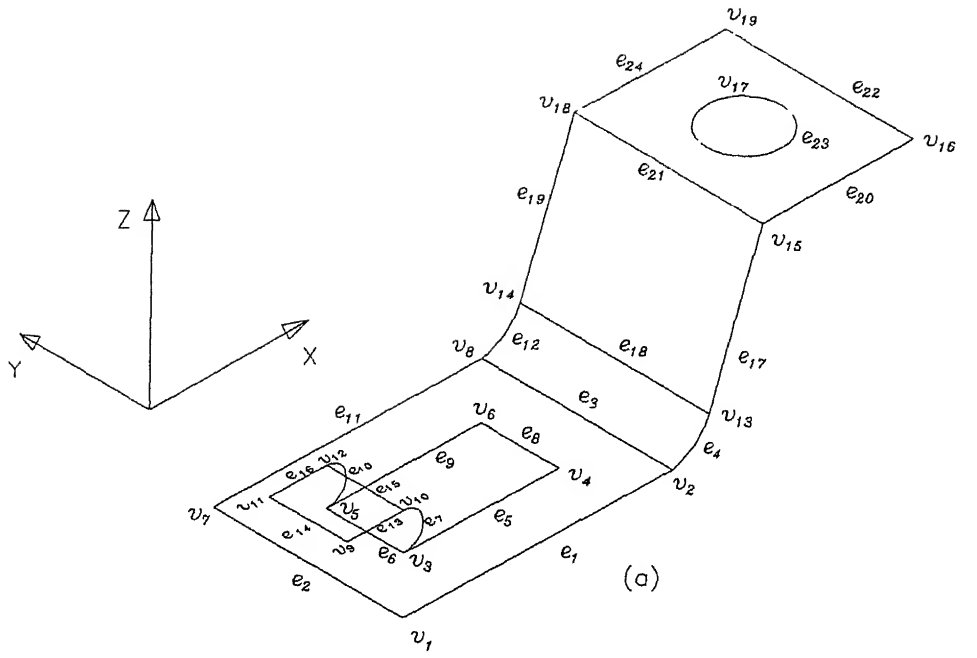
$$a_{i,i'} = \begin{cases} 1 & \text{if } v_i \in e_m, \text{ and } v_i' \in e_m \\ 0 & \text{otherwise} \end{cases}$$

$$i \in I, i' \in I, \text{ and } i' \neq i \quad \forall\ m \in M \tag{3.11}$$

where $a_{i,i'}$ indicates the value of i^{th} row and i'^{th} column of the VA matrix.

Implementation

During implementation of fusion methodology, discarding the specified row and column of the fusing vertex is difficult and time consuming. Therefore, rows and columns are



$v_{i'}$ / v_i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	–	1					1												
2	1	–						1					1						
3			–	1	1					1									
4			1	–		1													
5			1		–	1						1							
6				1	1	–													
7	1						–	1											
8		1					1	–						1					
9									–	1	1								
10			1						1	–		1							
11									1		–	1							
12					1					1	1	–							
13		1											–	1	1				
14								1					1	–				1	
15													1		–	1		1	
16															1	–			1
17																	–		
18														1	1			–	1
19																1		1	–

(b)

Figure 3.10· (a) Graph of the 3–D component with vertices and edges labeled according t the lexicographical order, and (b) Vertex adjacency matrix for the component graph.

ordered vertex set V The $d(v_i)$ is determined from the VA matrix as stated below .

$$d(v_i) = \sum_{\forall i} a_{i,i'} \quad \forall \quad i' \tag{3 12}$$

If $d(v_i)$ is $\begin{cases} \leq 2, & \text{then the graph is for 2–D component/part layout} \\ otherwise & \text{the graph can be either for 2–D or for 3–D component} \end{cases}$

$$\forall i \in I \tag{3 13}$$

If $d(v_i) > 2$, then plane equations are determined at the vertex v_i by vector normal method, with the help of edges incident at the vertex.

Let $e_{m'}$, $e_{m''}$ and $e_{m'''}$ be three edges incident at the vertex v_i, whose $d(v_i)$ is '3'. The plane equations are determined by finding vectors along the edges The vectors are given by

$$\vec{V}_{i,m'} = (v_i^x - v_{i'}^x)\vec{i} + (v_i^y - v_{i'}^y)\vec{j} + (v_i^z - v_{i'}^z)\vec{k} \tag{3.14(a)}$$

$$\vec{V}_{i,m''} = (v_i^x - v_{i''}^x)\vec{i} + (v_i^y - v_{i''}^y)\vec{j} + (v_i^z - v_{i''}^z)\vec{k} \tag{3 14(b)}$$

$$\vec{V}_{i,m'''} = (v_i^x - v_{i'''}^x)\vec{i} + (v_i^y - v_{i'''}^y)\vec{j} + (v_i^z - v_{i'''}^z)\vec{k} \tag{3 14(c)}$$

where $v_{i'}$, $v_{i''}$, and $v_{i'''}$ are other vertices of the three edges $e_{m'}$, $e_{m''}$, and $e_{m'''}$ respectively.

For example, in Fig. 3.11(a), $d(v_3)$ and $d(v_5)$ are '3', and the start and end vertices of edges (e_4, e_5, e_6, e_8, e_9, and e_{10}) incident on vertices v_3 and v_5 are given below :

$e_4 = \{v_3, v_4\}$, $e_5 = \{v_3, v_7\}$, $e_6 = \{v_3, v_{11}\}$, $e_8 = \{v_5, v_6\}$, $e_9 = \{v_5, v_9\}$, and

$e_{10} = \{v_5, v_{13}\}$

In Fig. 3.11(b), there are six vertices (v_5, .., v_10), having degree as '3'. The start and end vertices of edges incident on the vertex, say v_5 is given below

$e_2 = \{v_1, v_5\}$, $e_7 = \{v_5, v_6\}$, $e_8 = \{v_5, v_7\}$.

Similarly, edges incident on other vertices can be obtained from the Fig 3.11(b)

The edge vectors determined from eqn. (3 14) can lie in the same plane or in different planes. This is determined by finding the normal vectors ($\vec{NV}$) to the edge vectors ($\vec{V}_{i,m'}$, $\vec{V}_{i,m''}$, and $\vec{V}_{i,m'''}$) by cross product, with reference to one of the edge vectors Let $\vec{V}_{i,m'}$ be the reference vector Then normal vectors $\vec{NV}_{m',m''}$, and $\vec{NV}_{m',m'''}$ are given by :

$$\vec{NV}_{m',m''} = \vec{V}_{i,m'} \times \vec{V}_{i,m''}, \tag{3.15(a)}$$

$$\vec{NV}_{m',m'''} = \vec{V}_{i,m'} \times \vec{V}_{i,m'''}. \tag{3 15(b)}$$

and plane equations are given by

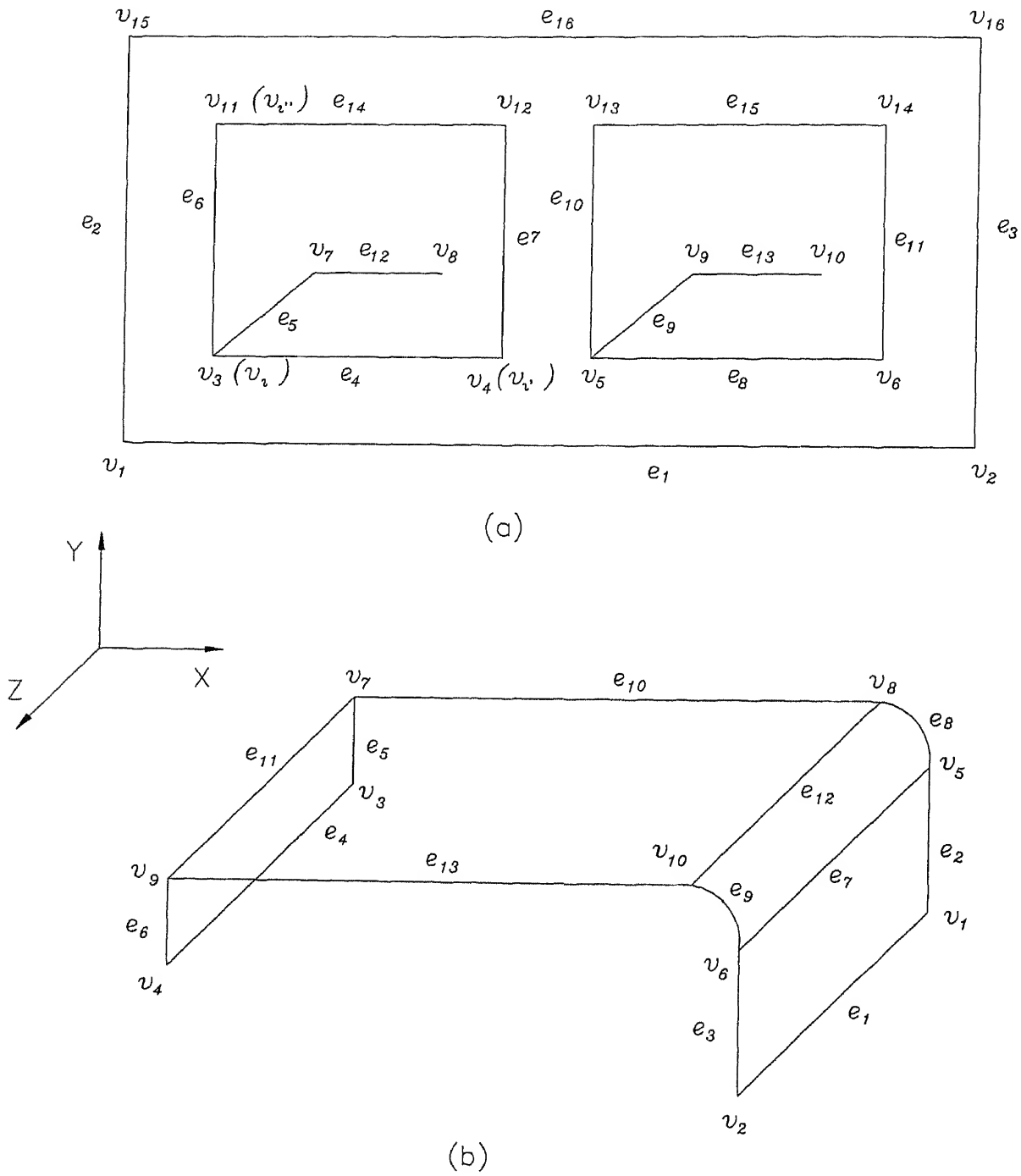


Figure 3 11· (a) A 2–D component nested layout graph, and (b) A 3–D component graph

$$NV^{x}_{m',m''}x + NV^{y}_{m',m''}y + NV^{z}_{m',m''}z = CV_{m',m''} \qquad (3\ 16(a))$$

$$NV^{x}_{m',m'''}x + NV^{y}_{m',m'''}y + NV^{z}_{m',m'''}z = CV_{m',m'''} \qquad (3.16(b))$$

If the coefficients in plane eqns (3 16(a)) and (3 16(b)) are equal, then the edges lie in one plane only (i e , the graph represent 2–D component, Fig 3.11(a)), else they belong to different planes (i.e , the graph represent 3–D component, Fig 3.11(b)) The flowchart for identifying the nature of component/part layout is given in Fig. 3 12

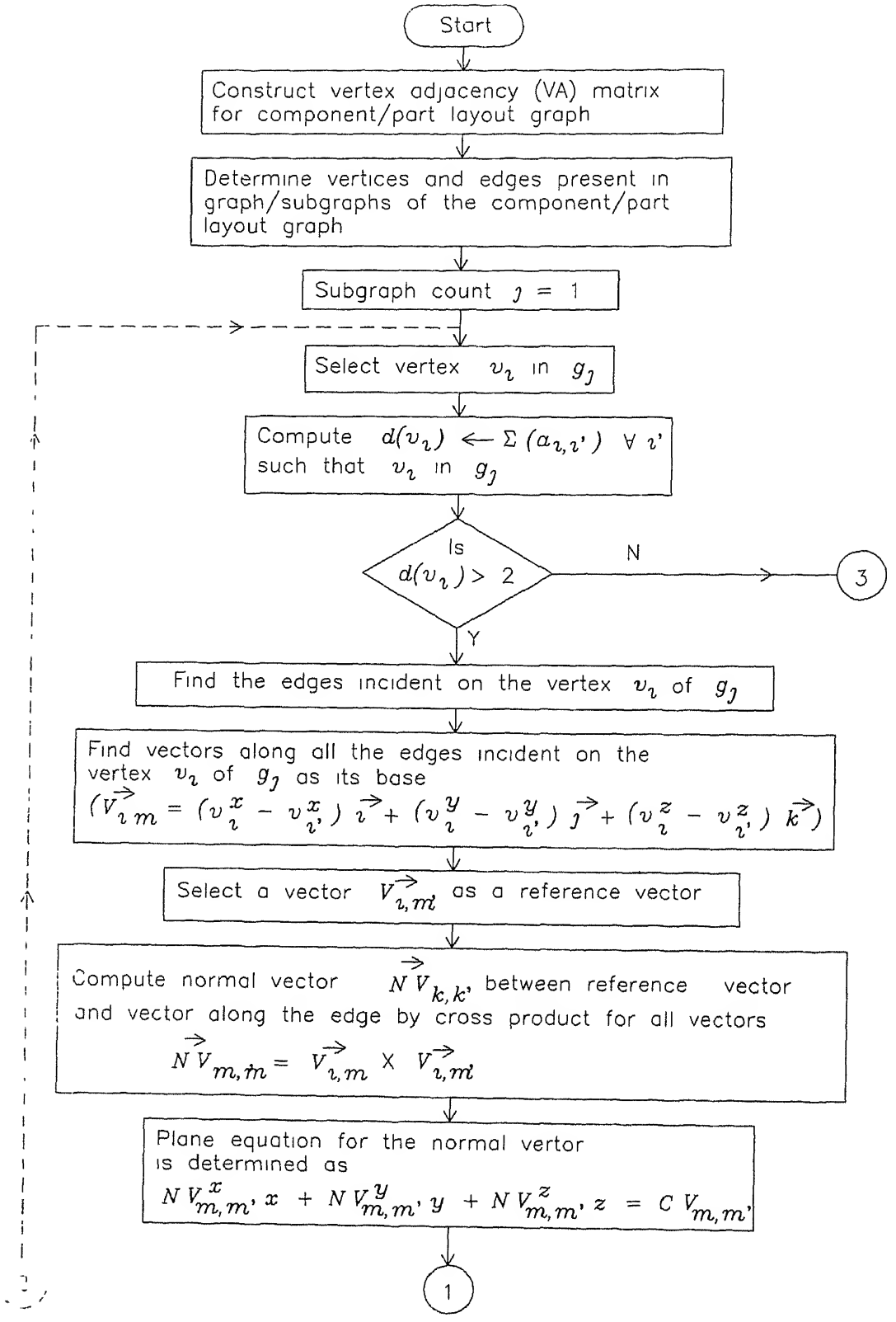


Figure 3.12· Flowchart for identifying the nature of the drawing given as input to the system

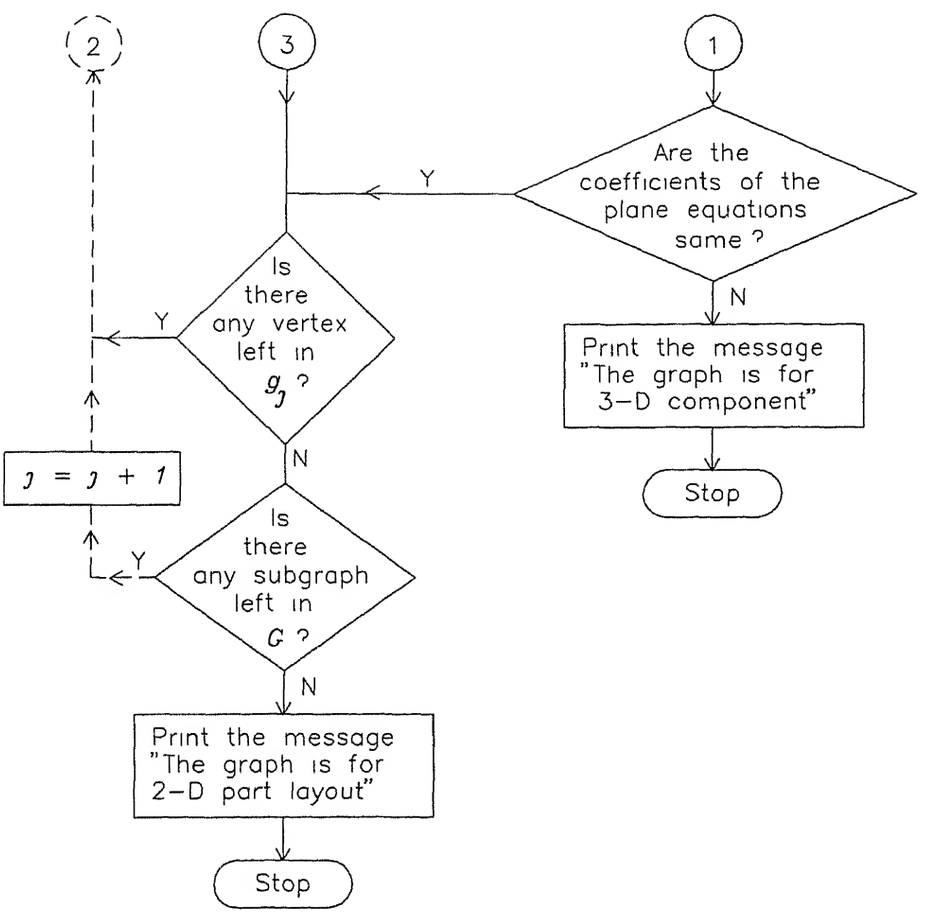


Figure 3 12 Flowchart for identifying the nature of the drawing given as input to the system (continued)

Chapter 4

Generation and Identification of Feature Sets

4.1 Introduction

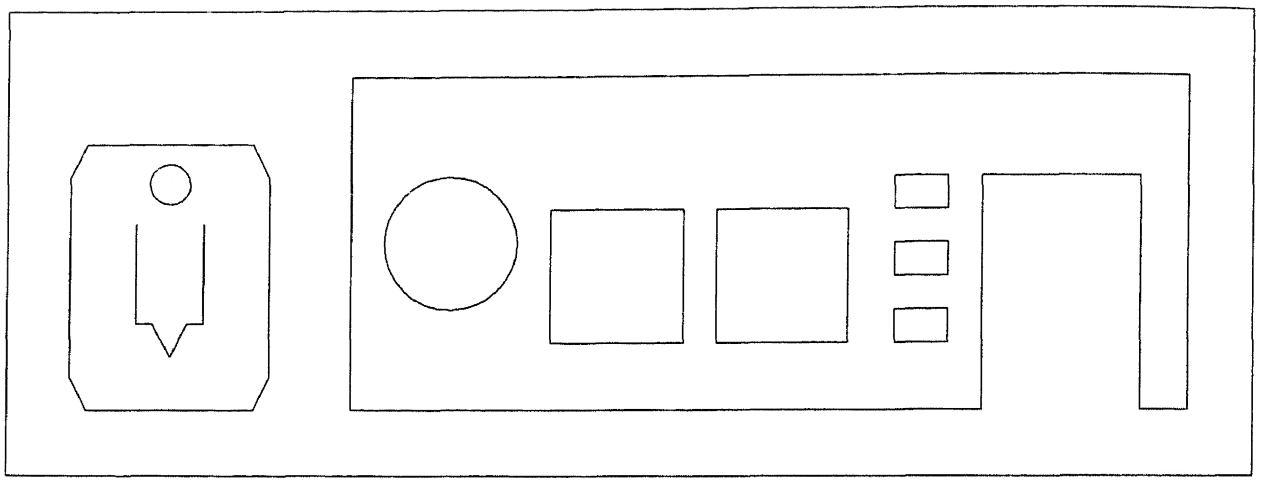
A set is a well defined list, collection or class of objects. The objects can be anything . numbers, people, letters, rivers, etc., (Lipschutz, 1981) In the present work, a set is defined as a collection of features, such that its 2–D geometrical shape can be produced by sheet metal pressworking operations (defined in Section 2 3) Various definitions of set theoretic concepts are given in Appendix C, which are not meant to be exhaustive but to provide common vocabulary to aid the clarity of discussion.

In this chapter, initially minimum enclosing rectangle (MER) for entity groups is determined. Using MER along with ray containment test, entity groups representing 2–D component layout are classified into three categories viz , raw material feature set, inside feature set, and boundary feature set. Subsequently, the component set and features in a component are identified.

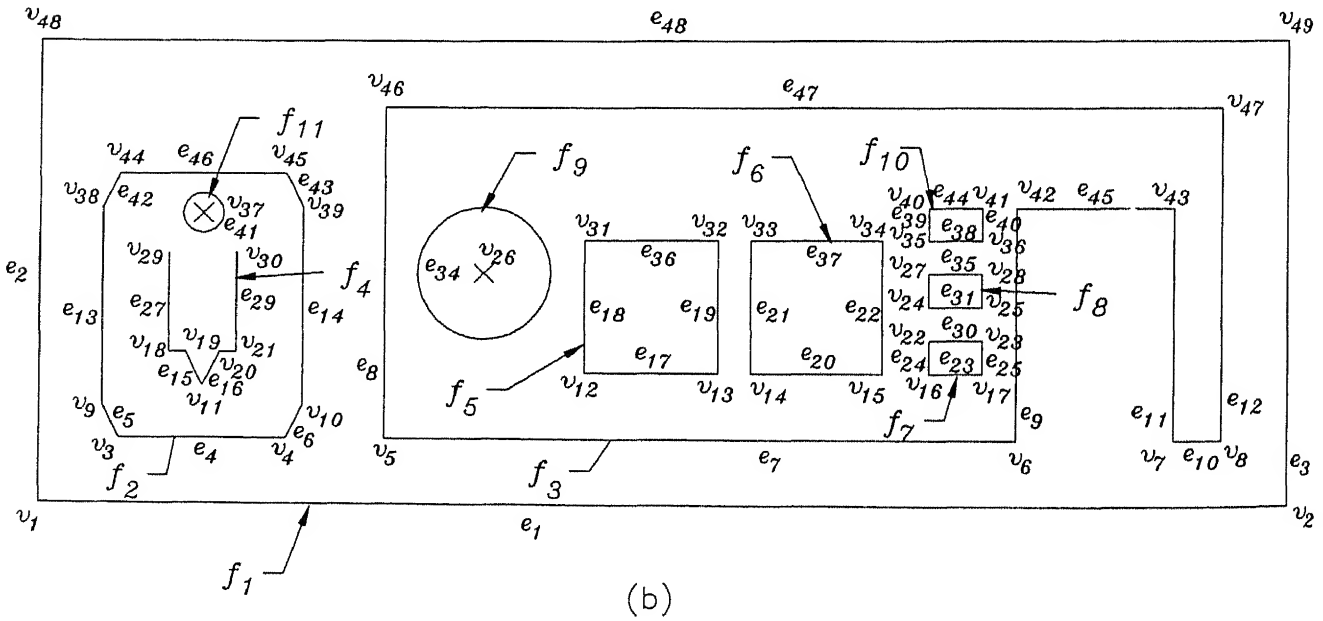
4.2 Classification of the feature set

Subgraphs of a component graph are identified (Section 3.6) and each subgraph represents a feature. For example, consider the component shown in Fig. 4.1(a), and its layout graph given in Fig 4.1(b) The vertices and edges, labelled in the graph are lexicographically ordered in accordance with the procedure discussed in Section 3.5, and the list of vertices is given in Table 4.1. The total number of subgraphs (features) identified in the above nested layout graph (Section 3.6) are eleven. Vertices belonging to each feature are given in Table 4 2.

The layout consists of entity groups (features) representing raw material, and components The components in turn have entity groups representing its boundary and internal



(a)



(b)

Figure 4 1: A 2–D dissimilar component nested layout (a) Nested layout, and (b) Nested layout graph

Vert No	x	y	rad /bul	start ang	end ang
1	13 0000	202 0000	–	–	–
2	200 0000	202 0000	–	–	–
3	25 0000	212 0000	–	–	–
4	50 0000	212 0000	–	–	–
5	65 0000	212 0000	–	–	–
6	160 0000	212 0000	–	–	–
7	183 0000	212 0000	–	–	–
8	190 0000	212 0000	–	–	–
9	23 0000	217 0000	–	–	–
10	53 0000	217 0000	–	–	–
11	38 0000	220 0000	–	–	–
12	95 0000	222 0000	–	–	–
13	115 0000	222 0000	–	–	–
14	120 0000	222 0000	–	–	–
15	140 0000	222 0000	–	–	–
16	147 0000	222 0000	–	–	–
17	155 0000	222.0000	–	–	–
18	33 0000	225 0000	–	–	–
19	35 0000	225 0000	–	–	–
20	40 0000	225.0000	–	–	–
21	43 0000	225 0000	–	–	–
22	147.0000	227 0000	–	–	–
23	155 0000	227 0000	–	–	–
24	147.0000	232 0000	–	–	–
25	155 0000	232 0000	–	–	–
26	80 0000	237 0000	10 0000	–	–
27	147.0000	237 0000	–	–	–
28	155 0000	237 0000	–	–	–
29	33 0000	240 0000	–	–	–
30	43.0000	240 0000	–	–	–
31	95 0000	242 0000	–	–	–
32	115 0000	242 0000	–	–	–
33	120 0000	242 0000	–	–	–
34	140 0000	242 0000	–	–	–
35	147 0000	242 0000	–	–	–
36	155 0000	242 0000	–	–	–
37	38 0000	246 0000	3 0000	–	–
38	23 0000	247 0000	–	–	–
39	53 0000	247 0000	–	–	–
40	147 0000	247 0000	–	–	–
41	155 0000	247 0000	–	–	–
42	160 0000	247 0000	–	–	–
43	183 0000	247 0000	–	–	–
44	25 0000	252 0000	–	–	–
45	50 0000	252 0000	–	–	–
46	65 0000	262 0000	–	–	–
47	190 0000	262 0000	–	–	–
48	13 0000	272 0000	–	–	–
49	200 0000	272.0000	–	–	–

Table 4 1· List of vertices after lexicographical ordering for component layout shown in Fig 4 1

Feature No	Vertex Nos and its degree			
1	1 (2)	2 (2)	48 (2)	49 (2)
2	3 (2)	4 (2)	9 (2)	10 (2)
	38 (2)	39 (2)	44 (2)	45 (2)
3	5 (2)	6 (2)	7 (2)	8 (2)
	42 (2)	43 (2)	46 (2)	47 (2)
4	11 (2)	18 (2)	19 (2)	20 (2)
	21 (2)	29 (1)	30 (1)	
5	12 (2)	13 (2)	31 (2)	32 (2)
6	14 (2)	15 (2)	33 (2)	34 (2)
7	16 (2)	17 (2)	22 (2)	23 (2)
8	24 (2)	25 (2)	27 (2)	28 (2)
9	26 (0)			
10	35 (2)	36 (2)	40 (2)	41 (2)
11	37 (0)			

Table 4 2 Vertices present in various feature sets

features (like pierced holes and lancing features). Let F be a feature set having its elements as raw material feature and features representing components. A component feature in turn consists of a boundary feature and inside features. The sets containing raw material feature and component features are represented by raw material feature set (R), and component set (C), respectively. Similarly, the sets of boundary features and inside features are denoted by boundary feature set (B), and inside feature set (N), respectively. Thus, the feature set F is given by

$$F = R \cup C, \tag{4.1}$$

and

$$C = B \cup N. \tag{4.2}$$

Therefore, feature set F is

$$F = R \cup B \cup N \tag{4 3}$$

The elements of various feature sets are determined in two levels :

1. In the first level, Minimum Enclosing Rectangle (MER) for each feature in the feature set is determined This is general in nature. MER is the minimum raw material required to manufacture a feature.

2. In the second level, geometry of the MER and ray containment test are used to identify the type of feature (i.e., raw material feature, boundary feature, or inside feature), and the component set. Ray containment test (Burger and Gillies, 1990) is a technique used to determine, whether an arbitrary point lies inside or outside the

given polygon, by means of the intersection of line segments. Thus, the second level is domain specific.

In the following section, determination of MER for features (i e. entity groups) is given

4.3 Minimum enclosing rectangle (MER) for features

Minimum enclosing rectangle (MER) for a feature is determined, such that its sides are parallel to the X and Y axes. Therefore, for any feature, its MER will have four vertices and four edges Vertices are determined by finding the maximum and minimum coordinate values of the feature along X and Y directions

The feature set F is also represented by

$$F = \{f_j \mid j \in J\}, \tag{4.4}$$

where f_j is j^{th} feature of the feature set F, and J is set of subscripts of features

Maximum and minimum x coordinate values for j^{th} feature in F are represented as $v^x_{max,j}$ and $v^x_{min,j}$ respectively Similarly, maximum and minimum y coordinate values of j^{th} feature in F are represented as $v^y_{max,j}$ and $v^y_{min,j}$, respectively

$$\begin{aligned} v^x_{max,j} &= \max_{v_i \in V_j} (v^x_i) \\ v^x_{min,j} &= \min_{v_i \in V_j} (v^x_i) \\ v^y_{max,j} &= \max_{v_i \in V_j} (v^y_i) \\ v^y_{min,j} &= \min_{v_i \in V_j} (v^y_i) \\ &\forall \quad j \in J \end{aligned} \tag{4.5}$$

where v^x_i and v^y_i represent x and y coordinate values, respectively of i^{th} vertex (v), and V_j represents vertex set of j^{th} element (feature) of F.

Thus, the four vertices of the MER for features are given by :

$$(v^x_{min,j}, v^y_{min,j}) \tag{4.6(a)}$$

$$(v^x_{min,j}, v^y_{max,j}) \tag{4.6(b)}$$

$$(v^x_{max,j}, v^y_{max,j}) \tag{4.6(c)}$$

$$(v^x_{max,j}, v^y_{min,j}) \tag{4.6(d)}$$

$$\forall \quad j \in J.$$

Feat No	v_1		v_2		v_3		v_4	
1	13 000	202 000	13 000	272 000	200 000	272 000	200 000	202 000
2	23 000	212 000	23 000	252 000	53 000	252 000	53 000	212 000
3	65 000	212 000	65 000	262 000	190 000	262 000	190 000	212 000
4	33 000	220 000	33 000	240 000	43 000	240 000	43 000	220 000
5	95 000	222 000	95 000	242 000	115 000	242 000	115 000	222 000
6	120 000	222 000	120 000	242 000	140 000	242 000	140 000	222 000
7	147 000	222 000	147 000	227 000	155 000	227 000	155 000	222 000
8	147 000	232 000	147 000	237 000	155 000	237 000	155 000	232 000
9	70 000	227 000	70 000	247 000	90 000	247 000	90 000	227 000
10	147 000	222 000	147 000	227 000	155 000	227 000	155 000	222 000
11	35 000	243 000	35 000	249 000	41 000	249 000	41 000	243 000

Table 4.3 Coordinates of vertices belonging to MER of features

Four edges of MER parallel to X and Y axes are given by the lines formed by joining the above equations in cyclic order. The coordinate values of vertices belonging to the MER for each feature of the layout shown in Fig 4.1 are given in Table 4 3, and they are determined using eqn. (4 5) MER and ray containment test are used to determine the type of features, and hence the various feature sets

In the following section ray containment test is explained to determine the containment of a feature inside the other feature.

4.4 Ray containment test for features

In ray containment test, the intersection of line segments is used to determine whether an arbitrary point P (Fig. 4.2) lies inside or outside the given polygon (Burger and Gillies, 1990) The mathematical model of the ray containment test for a polygon is given in Appendix D

The number of times a ray (i e. horizontal line) crosses the polygon edges is counted by solving the eqn. D 4. If the count is *odd*, then the point $P(x, y)$ is said to be contained and if it is *even*, then the point $P(x, y)$ is said to be outside the polygon

At procedural level, there are special cases that cause difficulties. A ray $\vec{HL}$ emerging from a point P may meet the edges of the polygon in any one of the ways represented in Fig. 4 3, i e , the ray may intersect the edge (Fig. 4.3(a)), pass along an edge (Fig. 4.3(b)), or pass through a vertex (Fig. 4.3(c)). To get the desired results, different numerical values are assigned to these cases. The numerical value assigned for *crossing an edge* is *one* (Fig. 4.3(a)). If the ray passes along an edge (e_2 in Fig. 4.3(b)), the other two edges (e_1 and e_3) which meet the edge (e_2) through which the ray passes, are analyzed. If these edges

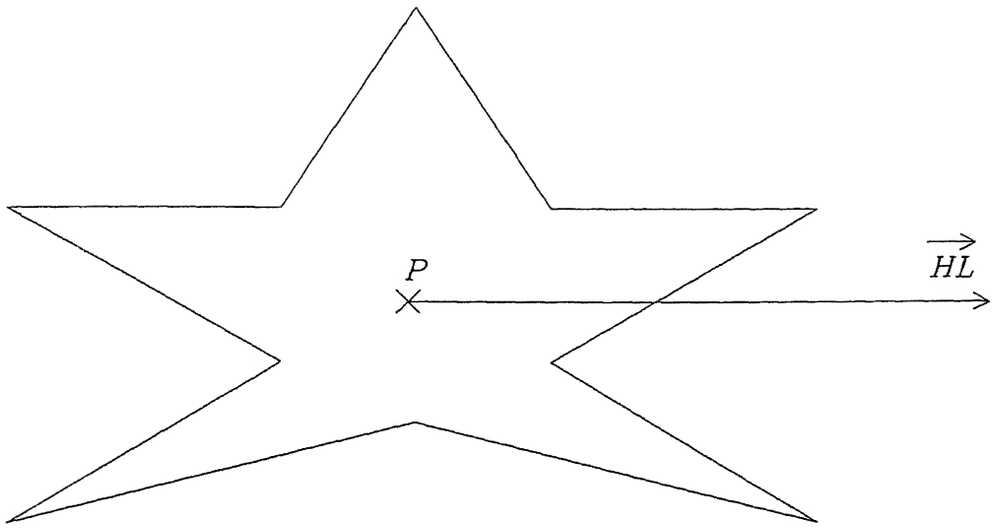


Figure 4.2: Ray containment test to determine whether a point (P) is contained in a polygon or not

(e_1 and e_3) lie on the opposite sides of the horizontal ray (Fig 4 3(b(i))), then a numerical value of *one* is assigned, and if they lie on the same side of the ray (Fig. 4.3(b(ii)) and Fig 4 3(b(iii)), then a numerical value of *two* is assigned. If the ray passes through a vertex (v_2 in Fig. 4.3(c)), and if two edges of a polygon meeting at the vertex under consideration lie on the opposite sides of the horizontal ray (e_1 and e_2 in Fig. 4.3(c(i))), then a numerical value of *one* is assigned. On the other hand, if the edges (e_1 and e_2) lie on the same side of the ray (Fig. 4.3(c(ii)) and Fig 4.3(c(iii))), then a numerical value of *two* is assigned. By taking these factors into consideration, a method for finding, whether a feature is inside or outside of the polygon is given below :
Let

$P_{j'}$ = duplet vertex, representing x and y coordinates belonging to a feature $f_{j'}$, whose containment has to be tested with other feature f_j

I_j = set of subscripts of vertices belonging to j^{th} feature

The ray containment test is carried out for the feature $f_{j'}$, such that the horizontal ray $\vec{HL}_{j'}$ passes through the vertex $P_{j'}$. The ray equation is given by :

$$\vec{HL}_{j'} = \vec{P}_{j'} + \mu\,\vec{i} \quad (\mu > 0) \tag{4.7}$$

and the line equation of edges of feature f_j is given by

$$\vec{L}_{ij} = \vec{V}_i + \nu\,(\vec{V}_{i+1} - \vec{V}_i) \qquad i \in I_j \tag{4.8}$$

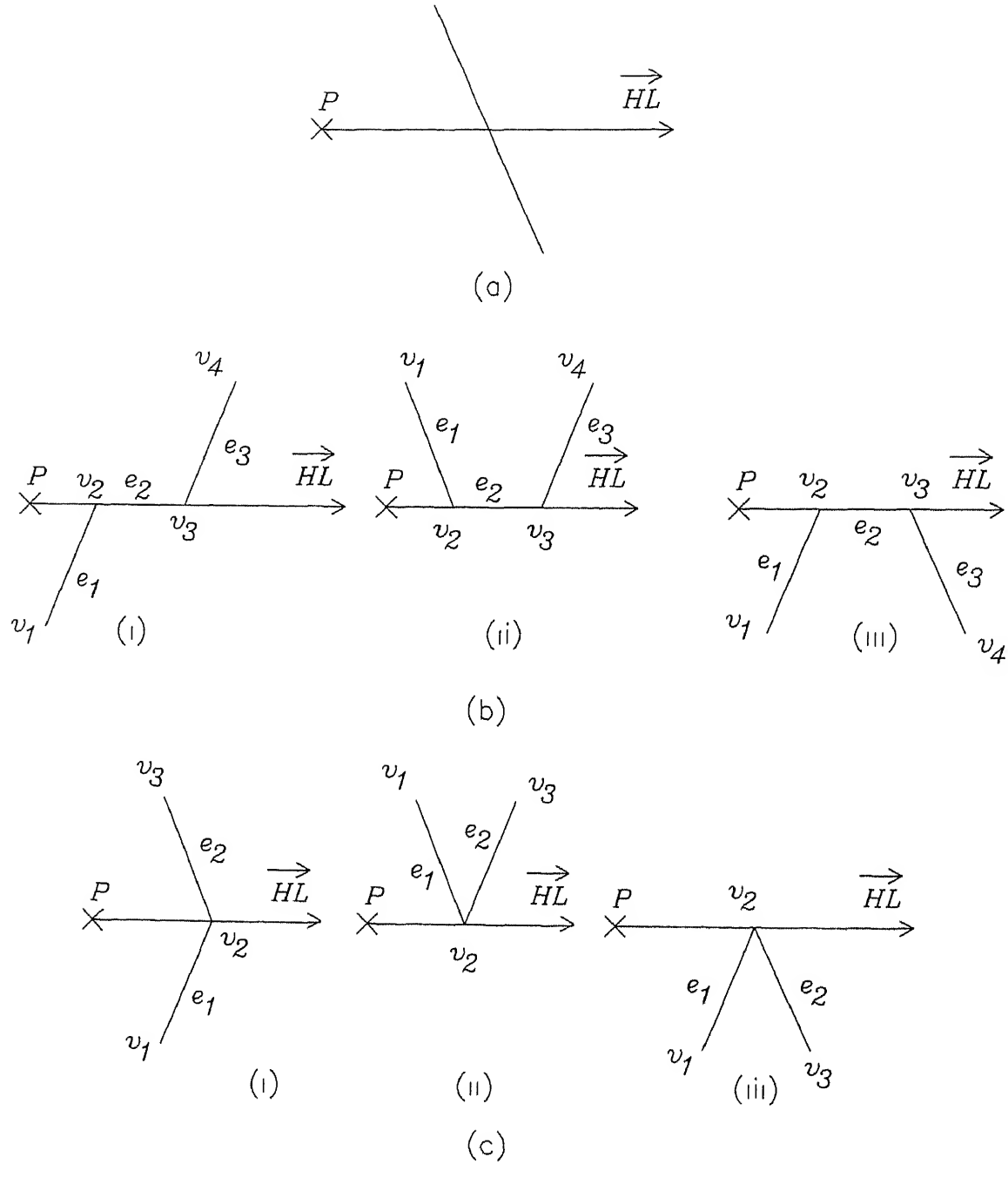


Figure 4.3. Various ways a horizontal ray intersects with edges of a polygon . (a) Intersecting an edge, (b) Passing along an edge, and (c) Passing through a vertex.

The intersection of the ray and line segments of the polygon is expressed by the vector equation

$$\vec{HL}_{j'} = \vec{L}_{ij} \tag{4 9}$$

i e ,

$$\vec{P}_{j'} + \mu\, \vec{i} = \vec{V}_i + \nu\,(\vec{V}_{i+1} - \vec{V}_i) \qquad i \in I_j \tag{4 10}$$

Above equation is solved for μ and ν, and

$$\text{If} \quad \mu > 0 \text{ and } 0 \leq \nu \leq 1 = \begin{cases} true, & \text{intersection occurs} \\ false, & \text{intersection does not occur} \end{cases} \tag{4.11}$$

Number of times the ray intersects the edges of the feature f_j under consideration is counted Let $Nc_{j'}$ represent the sum of the values assigned to the total number of intersections between the ray $\vec{HL}_{j'}$ emerging from the j'^{th} feature and the edges of the j^{th} feature

$$\text{If} \quad Nc_{j'} = \begin{cases} odd, & \text{feature } f_{j'} \text{ belongs to the feature } f_j \\ even, & \text{feature } f_{j'} \text{ does not belong to the feature } f_j. \end{cases} \tag{4.12}$$

In this way, a feature $f_{j'}$ belonging to (i.e., located inside) another feature f_j is determined uniquely.

The above ray containment test for features is explained with an example. Consider the features shown in Fig. 4 4. Let f_2 and f_3 be the features for which it is required to determine whether they are located inside the feature f_1 or not. Let a horizontal ray ($\vec{HL}_{f_2}$) start from the vertex v_6, which belongs to the feature f_2. This ray intersects the feature f_1 at vertices v_7 and v_8 (Fig. 4.4), which is determined by following the above procedure. The value is assigned to each intersection as per the rules given above, and the sum of the values assigned to the total number of intersections (Nc_{f_2}) is two. Therefore, feature f_2 is outside the feature f_1 in accordance to the eqn (4.12). Similarly, consider a horizontal ray ($\vec{HL}_{f_3}$) starting from the vertex v_{15}, such that v_{15} belongs to the feature f_3. The ray passes along an edge whose vertices are v_{16} and v_{17} in feature f_1. Now, the the sum of the values assigned to the total number of intersections (Nc_{f_3}) counted as per the above rule is one Therefore, feature f_3 is inside the feature f_1 in accordance to the eqn (4 12). Thus whether a feature is contained in another feature or not is determined uniquely

In the following section, procedure for the identification of features is discussed. In this procedure, both MER and ray containment test are used.

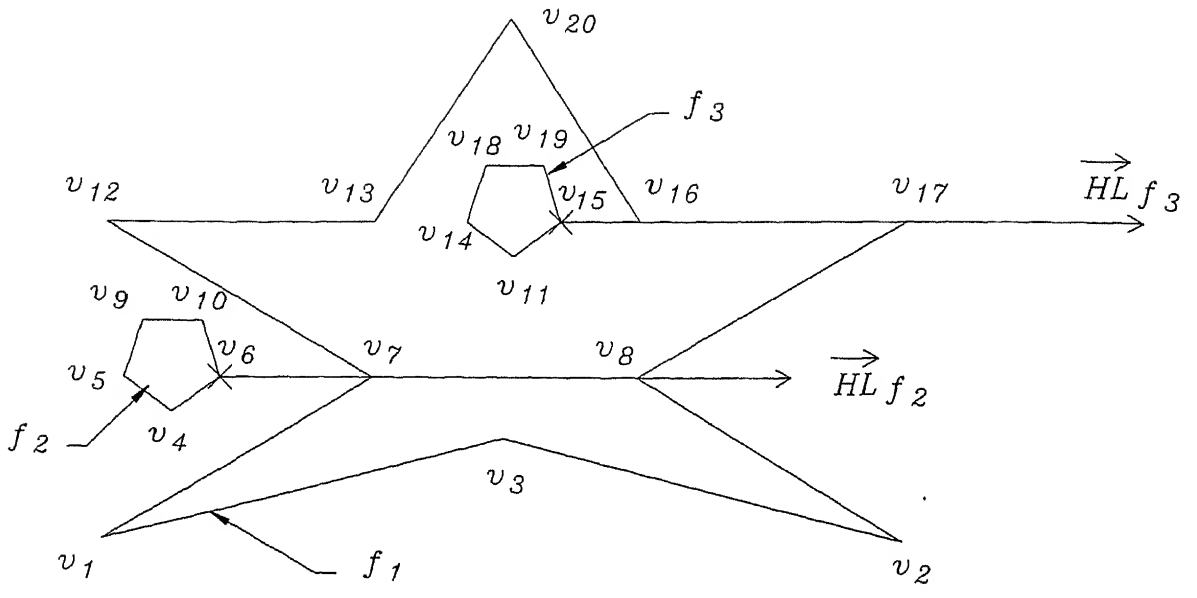


Figure 4.4· Ray containment test to determine whether a feature is contained in another feature or not.

4.5 Identification of various feature sets

The entities in the wireframe modeler are created randomly in a component layout, according to the assumptions stated in Section 3.2. Due to this, even after identifying the entity groups, it is not possible to obtain direct information from the component layout about the type of feature (i e. raw material feature, boundary feature, or inside feature) it is made of Hence, for every feature its type has to be determined from the feature set F. Therefore, a mathematical model has been developed for identifying the feature type from the set F, by relating the features geometrically and topologically. Following notations are used in the development of the procedure for identification of various feature sets.

b	=	element of the boundary feature set.
B	=	boundary feature set.
f_j	=	j^{th} element of the feature set.
F	=	feature set.
i	=	subscript of a vertex
I_j	=	j^{th} set of subscripts for the vertices of an entity group.
j	=	subscript of an entity group.
J	=	set of subscripts of the feature set elements.
k	=	subscript of a component set.
K	=	set of subscripts of the component set elements.
len	=	projected length along the axis.

max	=	subscript for the maximum value
min	=	subscript for the minimum value
n	=	element of the inside feature set
N	=	inside feature set
r	=	element of the raw material feature set
R	=	raw material feature set
v_i	=	i^{th} vertex of the vertex set
V	=	vertex set
x	=	superscript for the x coordinate value
y	=	superscript for the y coordinate value.

In eqn (4.3), R, B and N are partitions of F, i e.,

$$1 \text{ either } R = B \text{ or } R \cap B = \emptyset \qquad (4.13(a))$$

$$2. \text{ either } B = N \text{ or } B \cap N = \emptyset \qquad (4.13(b))$$

$$3. \text{ either } N = R \text{ or } N \cap R = \emptyset \qquad (4.13(c))$$

For classifying the feature set elements into R, B and N, initially the projected lengths of every element along X and Y directions are determined as given below

$$\begin{aligned} len_j^x &= v_{max,j}^x - v_{min,j}^x \\ len_j^y &= v_{max,j}^y - v_{min,j}^y \\ &\forall \quad j \in J \end{aligned} \qquad (4.14)$$

where len_j^x and len_j^y represent projected lengths of j^{th} element of feature set F along X and Y axes, respectively. $v_{max,j}^x$ and $v_{min,j}^x$ represent maximum and minimum x coordinate values respectively, of duplet vertex V of j^{th} element of F. Similarly, $v_{max,j}^y$ and $v_{min,j}^y$ represent maximum and minimum y coordinate values respectively of the duplet vertex v of j^{th} element of F.

Maximum and minimum x and y coordinate values of an element of feature set F are given by :

$$\begin{aligned} v_{max,j}^x &= \max_{i \in I_j} (v_{ij}^x) \\ v_{min,j}^x &= \min_{i \in I_j} (v_{ij}^x) \\ v_{max,j}^y &= \max_{i \in I_j} (v_{ij}^y) \\ v_{min,j}^y &= \min_{i \in I_j} (v_{ij}^y) \\ &\forall \quad j \in J \end{aligned} \qquad (4.15)$$

where v_{ij}^x and v_{ij}^y represent x and y coordinate values respectively of the i^{th} vertex v of the j^{th} element of F. The set I_j in the eqn (4.15) represents the set of subscripts of the j^{th} feature

4.5.1 Identification of raw material feature set

Raw material feature is defined as a feature whose projected length is maximum in both the directions (X and Y) among all the features Therefore, topologically raw material feature should be the outer most feature It may be noted that the raw material feature set is a single element set Mathematically the process of identification of the raw material feature set can be stated as follows :

$$len_{max}^x = \max_{j \in J} (len_j^x) \tag{4.16(a)}$$

$$Z1X = \{f_j \mid len_j^x = len_{max}^x, j \in J\} \tag{4.16(b)}$$

where len_{max}^x is the maximum projected length along X axis, and $Z1X$ is the feature set whose element has the maximum projected length along X direction.

Similarly, the maximum projected length along Y axis among all the elements in F (i.e , len_{max}^y) is given by the following equation :

$$len_{max}^y = \max_{j \in J} (len_j^y) \tag{4.17(a)}$$

and

$$Z1Y = \{f_j \mid len_j^y = len_{max}^y, j \in J\} \tag{4.17(b)}$$

where $Z1Y$ is the feature set whose element has the maximum projected length along Y direction.

It may be noted that the elements of $Z1X$ and $Z1Y$ sets are same, i e.,

$$f_j \in (Z1X \text{ and } Z1Y). \tag{4.18}$$

Therefore, f_j in the above equation is said to be a raw material feature, and the raw material feature set R is given by :

$$R = \{r \mid r = f_j, f_j \in Z1X \text{ or } f_j \in Z1Y, j \in J\}. \tag{4.19}$$

4.5.2 Identification of boundary and inside feature sets

Boundary feature of a component is the one located inside the raw material feature of a component layout, and may consist of features located inside it (inside features) Thus, the compliment set R' of raw material feature set R, is made up of boundary features and inside features Therefore, boundary feature set (B) and inside feature set (N) are partitions of R', and it is given by :

$$R' = B \cup N \tag{4 20}$$

such that

$$\text{either } B = N \text{ or } B \cap N = \emptyset \tag{4 21}$$

The set R' can also be stated as

$$R' = \{f_j \mid f_j \in F, f_j \notin R, j \in J\}. \tag{4 22}$$

Boundary and inside features are identified from set R' by the following algorithm :

Step 1. Find the minimum enclosing rectangle (MER) and its area for each element of R'

Step 2. Select a feature f_j, such that its MER has the maximum area in the complement set R' The feature f_j belongs to the boundary feature set B.

$$B = \{f_j \mid \text{area of MER of } f_j \text{ is maximum in } R'\} \tag{4.23}$$

Step 3. Determine whether the MER of the feature $f_{j'}$ present in the set R' (such that $f_{j'}$ is not equal to f_j) is located inside the MER of feature f_j under consideration or not.

Step 4. If MER of $f_{j'}$ is not inside the MER of f_j, then goto Step 6.

Step 5. MER of $f_{j'}$ is located inside the MER of f_j By ray containment test, determine whether the feature f'_j is inside the feature f_j or not

If the feature $f_{j'}$ is inside the feature f_j, then $f_{j'}$ is an inside feature and belongs to the inside feature set N. Discard this feature from the R' for further consideration in the identification process of elements of B and N.

$$N = \{f_{j'} \mid f_{j'} \notin B, f_{j'} \in R'\} \tag{4 24}$$

Step 6. Select another feature $f_{j'}$ from R', and goto Step 3. The process is repeated till all the features are exhausted from the complimentary feature set R'.

Step 7. Discard the feature f_j from the complimentary set R', since it is identified as a boundary feature. Goto Step 2. The process is repeated till all the elements of the R' are exhausted, and then stop.

It should be noted that the complement of raw material feature set R' is in the lexicographical order Therefore, sets B and N are also in the lexicographical order as these sets (B and N) are subsets of R'.

4.5.3 Implementation

During implementation, features belonging to the feature set F are not discarded from it after identification, but the flag attached to each feature is changed, and is not considered for further identification process. For every feature belonging to the inside feature set N, information about boundary feature in which it is located is also stored

The part layout shown in Fig 4.1(b) consists of eleven features (f_1, . ., f_{11}). Following the procedure of identification of features, feature f_1 is identified as an element of raw material feature set (R), since its projected lengths in both X and Y axes are maximum among all the features present in the feature set F. From the complement set R', MER of the feature f_3 has the maximum area, and it is identified as an element of boundary feature set (B) The flag attached to the feature f_3 is changed Therefore, f_3 is not considered further for identification process Other features are tested for the presence of their MER in the MER of boundary feature f_3. It is found out that MER of features f_5, f_6, f_7, f_8, f_9 and f_{10} are located inside the MER of f_3, and by ray containment test it is determined that features f_5 to f_9 are located inside the feature f_3. Therefore, features f_5 to f_9 are elements of inside feature set (N), and they belong to the boundary feature f_3. Flags of these features are also changed. Similarly, from the remaining elements in the set R', it is found that f_2 has maximum area, and features f_4 and f_{11} are located inside the boundary feature f_2. Therefore, feature f_2 belong to set B, and features f_4 and f_{11} belong to set N. Since the determination of the type of all features is complete, the identification process is stopped. The output of the proposed methodology for identification process of feature type is given in Table 4.4.

4.6 Identification of component set

Component set consists of components, where each element is in turn a set of boundary feature and inside features. Each element can have only one boundary feature, and any number of inside features. Therefore, boundary feature is said to be a *parent feature*, and inside features as *child features*, since they are analogous to the parent and children in a family.

In the previous section, features belonging to boundary feature set B and inside feature set N are identified. Also, parent feature of a child feature is determined. According to the

Feature set	F	$\{ f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, f_9, f_{10}, f_{11} \}$
Raw material feature set	R	$\{ f_1 \}$
Boundary feature set	B	$\{ f_2, f_3 \}$
Inside feature set	N	$\{ f_4, f_5, f_6, f_7, f_8, f_9, f_{10}, f_{11} \}$

Inside features belonging to a boundary feature

Boundary feature	f_2	f_4, f_{11}
Boundary feature	f_3	$f_5, f_6, f_7, f_8, f_9, f_{10}$

Table 4.4 **Various partition sets of the feature set, and inside features belonging to a boundary feature**

definition of a component, each element consists of only one boundary feature Therefore, the number of boundary features present in B will give the number of components present in the layout. Therefore, a component (c) is said to be made up of one boundary feature, and the component may consist any number of inside features located in it, and the set c is given by

$$c_k = \{f_j \mid \text{ either } f_j \in B \text{ or if } f_j \notin B, \text{ then it is inside } f_j \in \text{B}\} \tag{4.25}$$

and the component set is given by

$$C = \{c_k\} \tag{4.26}$$

where c_k is the k^{th} element of the component set C

As discussed above, elements of the component set and the features present in each element are determined for the layout given in Fig 4.1, and the output is given in Table 4.5 It should be noted that the cardinality of the component set is the same as that of the boundary feature set. For the present case, the cardinality of both B and C is two.

The component set identified is in lexicographical order, since the boundary feature set B is in lexicographical order, and components are identified in accordance with the features of the set B.

Component set	C	$\{ c_1, c_2 \}$
Component	c_1	$\{ f_2, f_4, f_{11} \}$
Component	c_2	$\{ f_3, f_5, f_6, f_7, f_8, f_9, f_{10} \}$

Table 4.5: **Component set, and features present in the component.**

Chapter 5

Shearing Feature Recognition System

5.1 Introduction

Shearing operations are very vital in industries manufacturing sheet metal components. Shearing operations produce 2–D form shapes without changing the nature of the component (i e., 2–D or 3–D) Various kinds of shearing operations used in different types of industries are given in Fig 2 2

The current literature (Section 1.3.2) on shearing operations shows that, most of the researchers have developed systems to identify only a few of the shearing operations listed in Fig. 2.2 since the concept of raw material type, layout type, and position of components in the layout are not considered by them. Knowledge of these aspects is essential in the process of identification of certain types of shearing operations Furthermore, the above systems can recognize shearing features from layouts having only similar type of components. With this in view, in the present work shearing features are identified from the optimum nested layout created by wireframe modeler for both *similar and dissimilar component layouts.* The standard DXF output file of AutoCAD Release-10 is used as input to the present system

In this chapter, the components identified in the previous chapter are further analyzed for identifying shearing features. Initially, position and direction of a component are determined with the help of geometry and cartesian octants in the given nested layout of components Then, adjacency relationship between components in the layout is determined by developing a set of principles. With the help of raw material feature type and adjacency component directional relationship (ACDR) table, the type of raw material, layout type, and direction of layout are determined. Subsequently, recognition process for various shearing features is discussed.

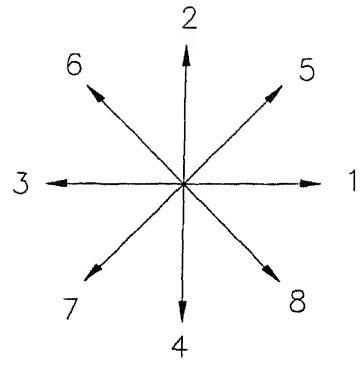


Figure 5 1: Adjacency directions in accordance to cartesian octants

5.2 Generation of adjacency component directional relationship (ACDR) table

The component set identified in the previous chapter is lexicographically ordered. This is not sufficient for identifying shearing features. To recognize these features, direction of components in the nested layout with respect to each other, their adjacency relationship, layout type, and direction of layout are also required. Therefore in the following sections, these information are determined. Subsequently ACDR table is generated.

5.2.1 Determination of direction for a component with respect to other components

The directional relationship of a component with respect to other components is determined in eight different directions according to the *cartesian octants* (Fig 5.1), and the directional set (D) (cartesian octants) is given by

$$D = \{d \mid d = 1, . \ , 8\} \tag{5.1}$$

where d is the direction of a cartesian octant in the directional set D.

Since, components have both length and width, it is difficult to determine the exact direction with respect to each other. Hence a heuristic procedure is developed to identify the direction.

In this heuristic procedure, a vertex from the vertex set (V_j) of j^{th} boundary feature is selected, such that it is the first vertex in the set of the lexicographically ordered set of V_j. Thus, this vertex precedes all other vertices in V_j, provided the boundary feature consists of only line type of entities. This vertex is considered as the head vertex of the component c_k. The direction of components between any two, with respect to each other is determined

Direction	Direction of $\vec{i}$	Direction of $\vec{j}$
1	+	0
2	0	+
3	−	0
4	0	−
5	+	+
6	−	+
7	−	−
8	+	−

Table 5 1 Rules for identifying the direction of an adjacent component c'_k with respect to the base component c_k

by the directional vector obtained between the head vertices of the two components. The directional vector of $c_{k'}$ component with respect to c_k is represented as $\vec{D}_{kk'}$, and is given by

$$\vec{D}_{kk'} = (h^x_{k'} - h^x_k)\,\vec{i} + (h^y_{k'} - h^y_k)\,\vec{j} \tag{5.2}$$

Based on the coefficients of the unit vectors ($\vec{i}$ and $\vec{j}$) in eqn (5 2), the direction of the component $c_{k'}$ w r.t c_k is determined as per the rules given in Table 5.1 (the layout is assumed to be in the XY plane).

Consider the layout shown in Fig 4.1 which has two dissimilar components. The boundary features of components c_1 and c_2 are f_2, and f_3, respectively as determined in Section 4 6 Vertices v_3, and v_5 precede all other vertices in the vertex set V_2, and V_3 Therefore they are identified as head vertices h_1 and h_2 for the components c_1 and c_2 respectively. The directional vector $\vec{D}_{1,2}$ (i.e., direction of c_2 w r.t. c_1) is determined according to eqn. (5 2), i e.,

$$\vec{D}_{1,2} = 40\,\vec{i} + 0\,\vec{j}.$$

Since the coefficient value of $\vec{i}$ is positive and the coefficient value of $\vec{j}$ is zero, therefore the adjacent component c_2 is in the direction '1' w.r t base component c_1 (Table 5.1) Similarly, the coefficient value of $\vec{i}$ is negative and the coefficient value of $\vec{j}$ is zero in $\vec{D}_{2,1}$, when c_1 is tested for the adjacency w.r.t. the base component c_2 Therefore, c_1 is in the direction '3' w.r t. c_2

It should be noted that, when arc type of entities are present in the boundary feature of a component, then pseudo vertex set (V'_j) (V'_j consists of pseudo vertices and the vertices of the vertex set V_j, such that V'_j is lexicographically ordered) of the feature is considered.

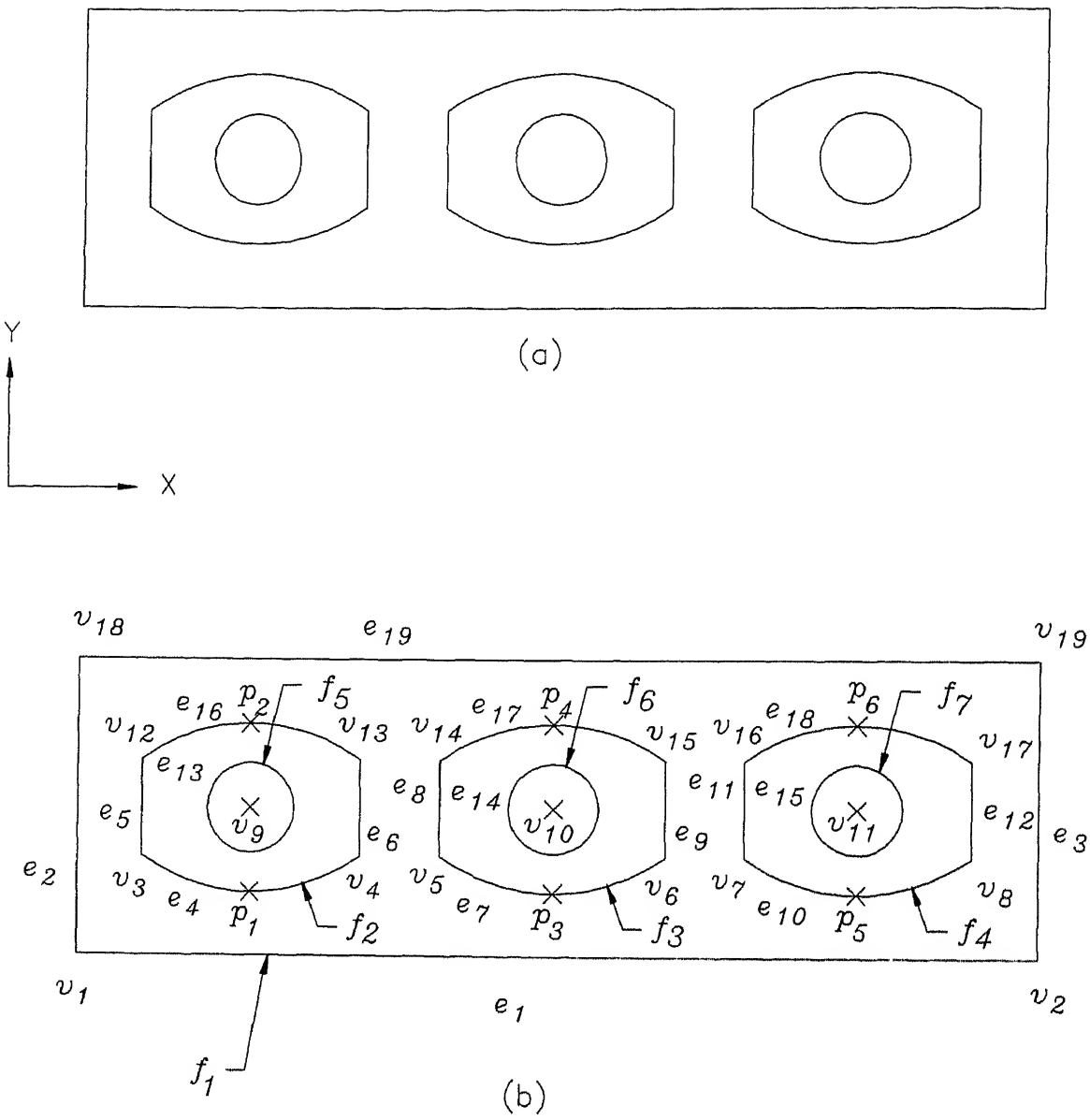


Figure 5 2: Component nested layout graph . (a) Layout of a 2–D component, and (b) Component layout graph representing pseudo vertices and features.

From this vertex set, the vertex which precedes all other vertices for the feature is chosen as head vertex.

For example, consider the component layout shown in Fig. 5.2(a). Its graph is given in Fig. 5.2(b). A list of lexicographically ordered vertices is given in Table 5.2. Vertices present in each feature, and features present in various feature sets are given in Table 5.3 and Table 5.4 respectively. Since the boundary features f_2, f_3, and f_4 (Fig. 5.2(b)) consist

Vert No	x	y	rad./bul.	start ang	end ang.
1	30 0000	205 0000	–	–	–
2	275.0000	205.0000	–	–	–
3	40 0004	220 4997	–	–	–
4	75 0002	220.5001	30 0000	234.3150	305.6850
5	87 5004	220 4997	–	–	–
6	122.5002	220.5001	30.0000	234 3150	305 6850
7	135 0004	220 4997	–	–	–
8	170.0002	220 5001	30.0000	234.3150	305 6850
9	57 5000	228 0000	7.0000	–	–
10	105 0000	228 0000	7 0000	–	–
11	152 5000	228 0000	7.0000	–	–
12	40 0000	235 5000	30 0000	54 3150	125 6850
13	75 0000	235 5000	–	–	–
14	87 5000	235 5000	30 0000	54 3150	125 6850
15	122.5000	235.5000	–	–	–
16	135.0000	235.5000	30 0000	54 3150	125.6850
17	170 0000	235.5000	–	–	–
18	30.0000	251.0000	–	–	–
19	275 0000	251 0000	–	–	–

Table 5.2: List of vertices after lexicographical ordering

of arcs, the head vertex is determined from the pseudo vertex set V_j', where $(j = 2, 3, 4)$, of boundary features. The pseudo vertex is located on the entity arc, and is the point of intersection of arc and lines parallel to X or Y axis passing through the center of the arc. From the vertex set (V_j'), head vertex is chosen, such that it precedes all other vertices in V_j' set Thus the vertex set V_2' will have all the vertices present in V_2, and two pseudo vertices p_1, and p_2 as there are two arcs in the feature. From V_2', it is found out that p_1 precedes all other vertices, and hence p_1 is the head vertex for c_1. Similarly, p_3 and p_5 in Fig 5 2(b) represent the head vertices for c_2 and c_3, respectively. The coordinate values of pseudo vertices p_i $(i = 1, .. , 6)$ are given in Table. 5.5. In accordance to eqn.(5.2), the directional vectors $(\vec{D_{1,2}})$ and $(\vec{D_{1,3}})$ of c_2 and c_3, respectively have positive $\vec{i}$, and $\vec{j}$ is zero Therefore, the components c_2 and c_3 are in the direction of '1' w.r.t. c_1. Similarly, directions for other components can be determined

5.2.2 Identification of adjacent components

To determine the ACDR table, direction of a component w.r.t. another component and their adjacency should be known. Two components are said to be adjacent to one another,

Feature No.	Vertex Nos. and its degree			
1	1 (2)	2 (2)	18 (2)	19 (2)
2	3 (2)	4 (2)	12 (2)	13 (2)
3	5 (2)	6 (2)	14 (2)	15 (2)
4	7 (2)	8 (2)	16 (2)	17 (2)
5	9 (0)			
6	10 (0)			

Table 5 3: Vertices and their degree in the component graph

Feature set type			Feature numbers		
Raw material	(R)	:	1		
Boundary	(B)	:	5	6	7
Inside	(N)	:	2	3	4
Component	(C)	.	c_1	c_2	c_3
	c_1	:	2	5	
	c_2	:	3	6	
	c_3	:	4	7	

Table 5.4: Features present in various feature sets

Vert no.	x	y
1	57.5000	214.8670
2	57.5000	241.1830
3	105.0000	214.8670
4	105 0000	241 1830
5	152 5000	214.8670
6	152.5000	241.1830

Table 5.5. List of pseudo vertices present in boundary features.

Direction	Conditions
1	$v^x_{max,k} > v^x_{min,k'}$
2	$v^y_{max,k} < v^y_{min,k'}$
3	$v^x_{min,k} > v^x_{max,k'}$
4	$v^y_{min,k} < v^y_{max,k'}$
5	$v^x_{max,k} \geq v^x_{min,k'}$ and $v^y_{max,k} \geq v^y_{min,k'}$
6	$v^x_{min,k} \leq v^x_{max,k'}$ and $v^y_{max,k} \geq v^y_{min,k'}$
7	$v^x_{min,k} \leq v^x_{max,k'}$ and $v^y_{min,k} \leq v^y_{max,k'}$
8	$v^x_{max,k} \geq v^x_{min,k'}$ and $v^y_{min,k} \leq v^y_{max,k'}$

Table 5.6: Rules for determining the intersection between MER of components

if there is no other component present between them. To find the adjacency between any two components, a heuristic approach is employed, where the minimum enclosing rectangle (MER) of components (i.e. boundary features) are used. A component ($c_{k'}$) is said to be adjacent to the base component c_k, if the MER of c_k and MER of $c_{k'}$ do not have MER of any other components between them. If MER of components under consideration are intersecting, then they are said to be adjacent to each other. Rules for determining the intersection between MER of components are given in Table 5.6. If MER of components are not intersecting, then the minimum rectangle present between MER of the two components (MRBC) is determined. The rules for finding MRBC is given in Table 5.7. It should be noted that the MRBC is dependent on the direction of components.

Implementation

During implementation two components are considered to find their adjacency. If they are adjacent to each other, then a flag value of one is assigned, else zero is assigned. The adjacency relationship is with respect to each other. Therefore in vice-versa case adjacency relationship is not determined. This reduces the computational time. Also, if

Direction	Conditions	Vertices list
1	$v^x_{max,k} \leq v^x_{min,k'}$	$v_1 = (v^x_{max,k}, v^y_{min,k})$ $v_2 = (v^x_{min,k'}, v^y_{min,k'})$ $v_3 = (v^x_{max,k}, min(v^y_{max,k}, v^y_{max,k'}))$ $v_4 = (v^x_{min,k'}, min(v^y_{max,k}, v^y_{max,k'}))$
2	$v^y_{max,k} \leq v^y_{min,k'}$	$v_1 = (max(v^x_{min,k}, v^x_{min,k'}), v^y_{max,k})$ $v_2 = (min(v^x_{max,k}, v^x_{max,k'}), v^y_{max,k})$ $v_3 = (v^x_{min,k'}, v^y_{min,k'})$ $v_4 = (min(v^x_{max,k}, v^x_{max,k'}), v^y_{min,k'}))$
3	$v^x_{min,k} \geq v^x_{max,k'}$	$v_1 = (v^x_{max,k'}, v^y_{min,k'})$ $v_2 = (v^x_{min,k}, v^y_{min,k})$ $v_3 = (v^x_{max,k'}, min(v^y_{max,k}, v^y_{max,k'}))$ $v_4 = (v^x_{min,k}, min(v^y_{max,k}, v^y_{max,k'}))$
4	$v^y_{min,k} \geq v^y_{max,k'}$	$v_1 = (max(v^x_{min,k}, v^x_{min,k'}), v^y_{max,k'})$ $v_2 = (min(v^x_{max,k}, v^x_{max,k'}), v^y_{max,k'})$ $v_3 = (v^x_{min,k}, v^y_{min,k})$ $v_4 = (min(v^x_{max,k}, v^x_{max,k'}), v^y_{min,k}))$
5 (i)	$v^x_{max,k} < v^x_{min,k'}$ and $v^y_{max,k} > v^y_{min,k'}$	$v_1 = (v^x_{max,k}, v^y_{min,k'})$ $v_2 = (v^x_{min,k'}, v^y_{min,k'})$ $v_3 = (v^x_{max,k}, min(v^y_{max,k}, v^y_{max,k'}))$ $v_4 = (v^x_{min,k'}, min(v^y_{max,k}, v^y_{max,k'}))$
(ii)	$v^x_{max,k} \geq v^x_{min,k'}$ and $v^y_{max,k} \leq v^y_{min,k'}$	$v_1 = (max(v^x_{min,k}, v^x_{min,k'}), v^y_{max,k})$ $v_2 = (min(v^x_{max,k}, v^x_{max,k'}), v^y_{max,k})$ $v_3 = (max(v^x_{min,k}, v^x_{min,k'}), v^y_{min,k'})$ $v_4 = (min(v^x_{max,k}, (v^x_{max,k'}), v^y_{min,k'})$

Table 5.7. Rules for finding vertices of MRBC between components.

Direction	Conditions	Vertices list
6 (i)	$v^x_{min,k} > v^x_{max,k'}$ and $v^y_{max,k} > v^y_{min,k'}$	$v_1 = (v^x_{max,k'}, v^y_{min,k'})$ $v_2 = (v^x_{min,k}, v^y_{min,k'})$ $v_3 = (v^x_{max,k'}, min(v^y_{max,k}, v^y_{max,k'}))$ $v_4 = (v^x_{min,k}, min(v^y_{max,k}, v^y_{max,k'}))$
(ii)	$v^x_{min,k} \leq v^x_{max,k'}$ and $v^y_{max,k} \leq v^y_{min,k'}$	$v_1 = (max(v^x_{min,k}, v^x_{min,k'}), v^y_{max,k})$ $v_2 = (min(v^x_{max,k}, v^x_{max,k'}), v^y_{max,k})$ $v_3 = (max(v^x_{min,k}, v^x_{min,k'}), v^y_{min,k'})$ $v_4 = (min(v^x_{max,k}, (v^x_{max,k'}), v^y_{min,k'})$
7 (i)	$v^x_{min,k} > v^x_{max,k'}$ and $v^y_{min,k} < v^y_{max,k'}$	$v_1 = (v^x_{max,k'}, v^y_{min,k})$ $v_2 = (v^x_{min,k}, v^y_{min,k})$ $v_3 = (v^x_{max,k}, min(v^y_{max,k}, v^y_{max,k'}))$ $v_4 = (v^x_{min,k'}, min(v^y_{max,k}, v^y_{max,k'}))$
(ii)	$v^x_{min,k} \leq v^x_{max,k'}$ and $v^y_{min,k} \geq v^y_{max,k'}$	$v_1 = (max(v^x_{min,k}, v^x_{min\ k'}), v^y_{max,k'})$ $v_2 = (min(v^x_{max,k}, v^x_{max,k'}), v^y_{max,k'})$ $v_3 = (max(v^x_{min,k}, v^x_{min,k'}), v^y_{min,k})$ $v_4 = (min(v^x_{max,k}, (v^x_{max,k'}), v^y_{min,k})$
8 (i)	$v^x_{max,k} < v^x_{min,k'}$ and $v^y_{min,k} < v^y_{max,k'}$	$v_1 = (v^x_{max,k}, v^y_{min,k})$ $v_2 = (v^x_{min,k'}, v^y_{min,k})$ $v_3 = (v^x_{max,k}, min(v^y_{max,k}, v^y_{max,k'}))$ $v_4 = (v^x_{min,k'}, min(v^y_{max,k}, v^y_{max,k'}))$
(ii)	$v^x_{max,k} \geq v^x_{min,k'}$ and $v^y_{min,k} \geq v^y_{max,k'}$	$v_1 = (max(v^x_{min,k}, v^x_{min,k'}), v^y_{max,k'})$ $v_2 = (min(v^x_{max,k}, v^x_{max,k'}), v^y_{max,k'})$ $v_3 = (max(v^x_{min,k}, v^x_{min,k'}), v^y_{min,k})$ $v_4 = (min(v^x_{max,k}, (v^x_{max,k'}), v^y_{min,k})$

Table 5.7: Rules for finding vertices of MRBC between components (continued)

the component is adjacent to the base component, then only the component number and its direction are stored in the form of a linked list, else the information is not stored This helps in reducing the total memory requirement for the system A flowchart for determining the adjacency component relationship matrix, and the ACDR table is given in Fig. 5.3.

The output of the module for dissimilar components nested layout, shown in Fig. 5.4(a) is given in Fig. 5 5. The Fig. 5.4(c) gives features present in each component. Similarly, a component layout having intersecting MER of components are shown in Fig. 5 6(a) and its layout graph in Fig. 5.6(b). The ACDR table for this example is given in Fig. 5.6(c)

In the ACDR table (Fig. 5.5(b)), '1' in the hatched cell indicates that the components c_2 (say $c_{k'}$) is adjacent to c_1 (say c_k), and '5' in the right hand top corner of the hatched cell indicates the direction of the component c_2 w.r.t. c_1 (i.e., $c_{k'}$ w.r t. c_k), and is given by

$$a_{kk'} = \begin{cases} 1 & \text{if the component } c_{k'} \text{ is adjacent to } c_k \\ 0 & \text{otherwise} \end{cases}$$

$$\forall \quad c_k \neq c_{k'}$$

Directions of components ($d_{kk'}$) present in a layout with respect to the base component c_k is represented by the directional set D_k and is given by

$$D_k = \{d_{kk'} \mid a_{kk'} = 1,\ c_k \in C,\ c_{k'} \in C, \text{ and } c_k \neq c'_k\} \quad \forall \quad c_k \in C. \tag{5.3}$$

5.3 Identification of raw material type

Raw material types considered in the present research are coil, sheet and strip. Identification of raw material type helps in recognizing shearing features. The element (r) of the raw material feature set (R) is processed to identify its type.

The subgraph of the raw material feature (r) is of planar type, and either it will be circular (loop) or acyclic (path). If the subgraph is of acyclic type, then the raw material feature is identified as coil type. On the other hand, if the subgraph of r is of circular type, then the feature represents either sheet or strip. Distinction between sheet or strip is done from the geometry point of view (width by length ratio). The element r is said to of strip type, if the width by length ratio is less than 0.4, otherwise the feature is identified as sheet type.

For example, the subgraph of the element of R (Fig. 4.1) is of circular type, and the width by length ratio is 0.374, and hence the raw material is of strip type.

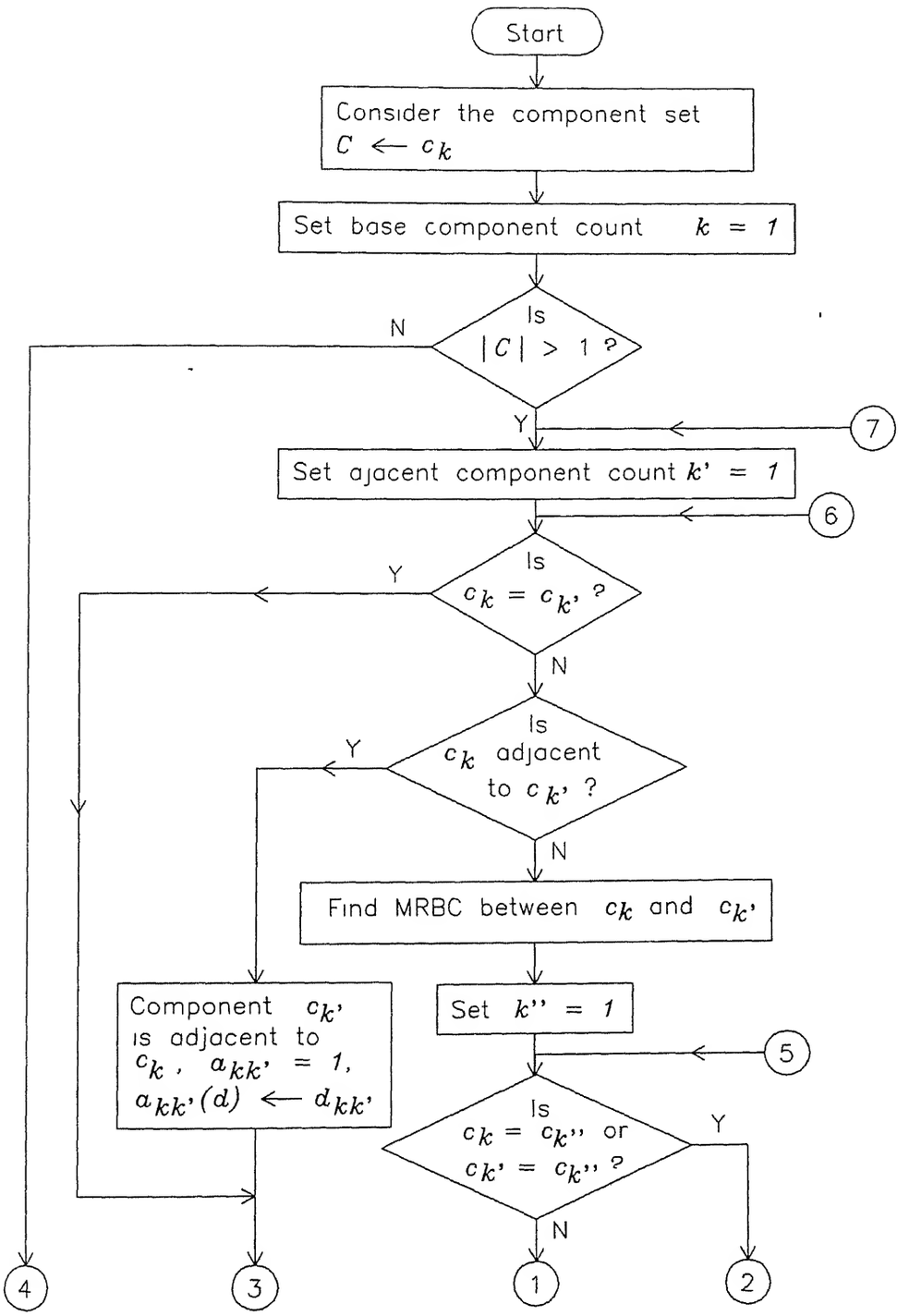


Figure 5 3: **Flowchart for determining adjacency component relationship matrix and ACDR table**

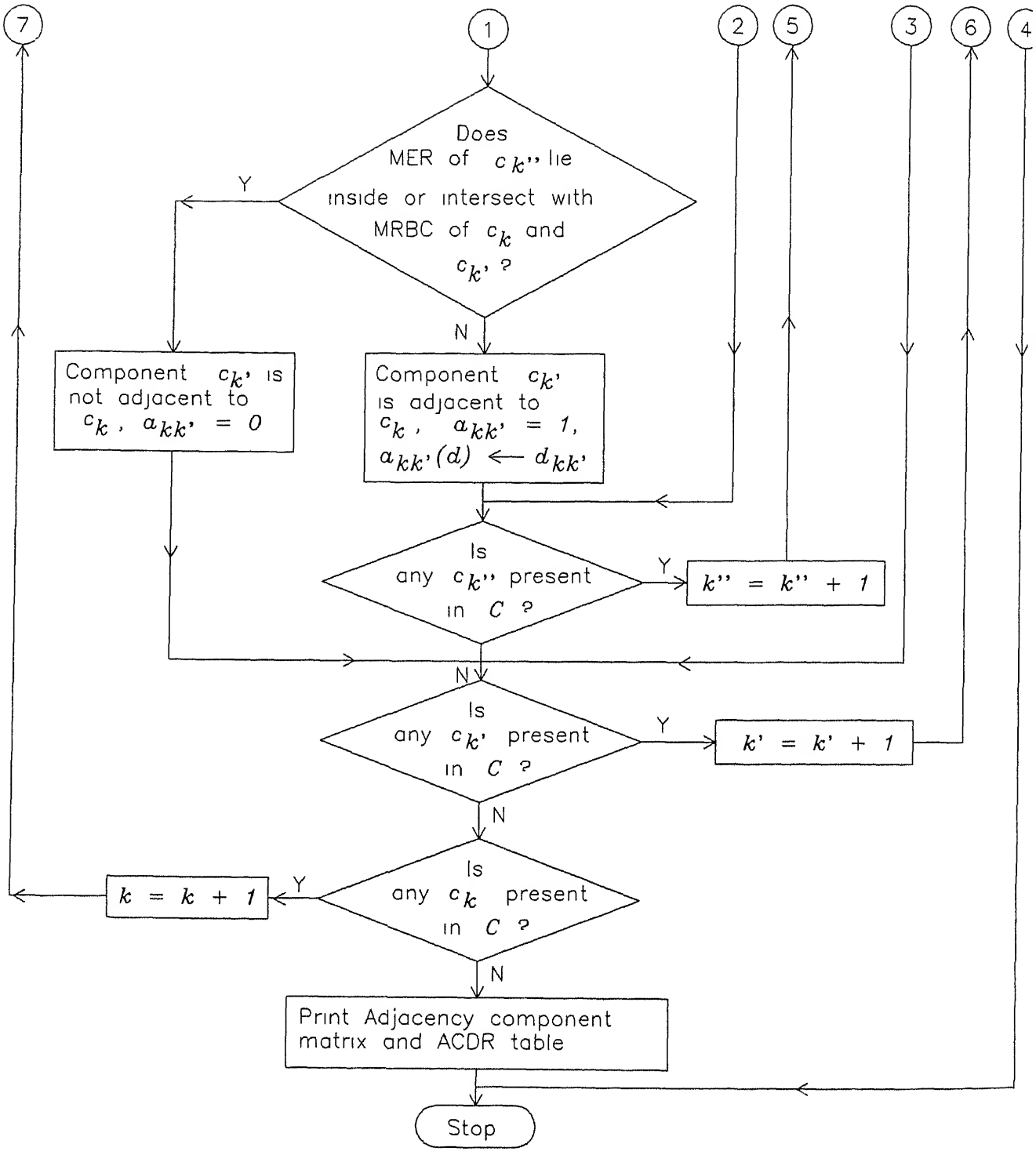
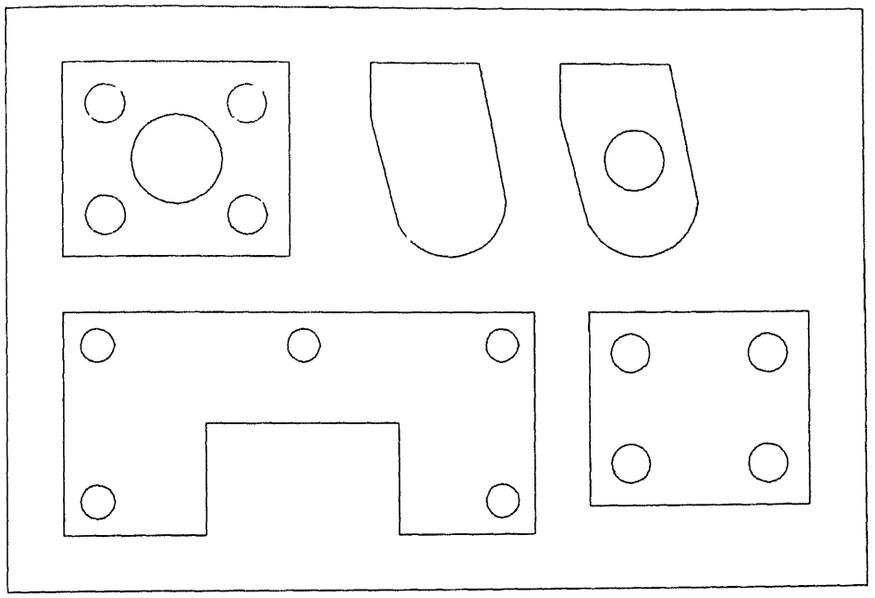
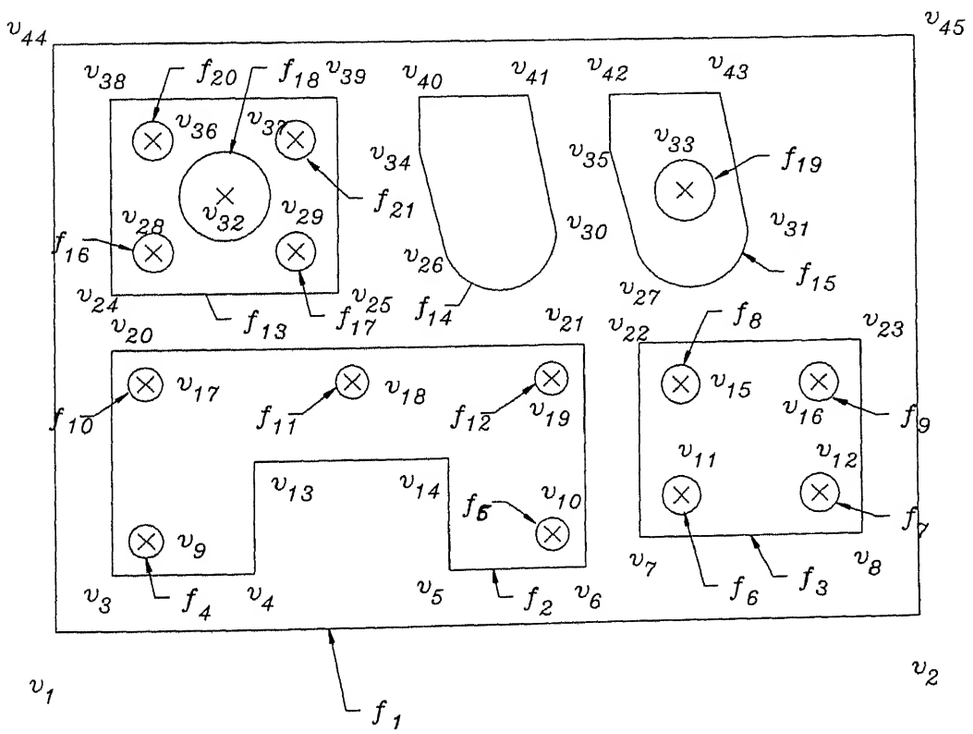


Figure 5.3: Flowchart for determining adjacency component relationship matrix and ACDR table (continued).



(a)



(b)

Figure 5.4: Dissimilar components nested layout · (a) Nested layout, (b) Layout graph, and (c) component set and features present in each component

Component set	C	c_1	c_2	c_3	c_4	c_5	
Features in	c_1	f_2	f_4	f_5	f_{10}	f_{11}	f_{12}
Features in	c_2	f_3	f_6	f_7	f_8	f_9	
Features in	c_3	f_{13}	f_{16}	f_{17}	f_{18}	f_{20}	f_{21}
Features in	c_4	f_{14}					
Features in	c_5	f_{15}	f_{19}				

(c)

Figure 5 4 Dissimilar components nested layout (a) Nested layout, (b) Layout graph, and (c) component set and features present in each component (continued)

5.4 Identification of layout type and the direction of layout

Shearing features like parting, slitting and shearing are dependent on the type of layout, which in turn depends on the number of rows and columns present in the nested layout. In the present work, layout type is broadly classified into three types viz., components present in rows only, in columns only, or in the mid of rows and columns. The methodology for determining the layout type and its direction is given below :

Methodology

Layout type is determined from the directional set of the layout (D_l), and the graph of the components present in different ~~the~~ directions. The set D_l is given by

$$D_l = \bigcup_{k \in K} D_k \tag{5.4}$$

where D_k is a set of directions of adjacent components of $c_{k'}$ w.r t. the base component c_k

Arrangement of the components in a layout (i e., whether they are in rows and/or in columns or in the mid of rows and columns) is determined by subgraphs of the components present in the directions '1' and '3', or '2' and '4'.

Let $G_{1,3}$ represent the subgraph of components having adjacent components present in the direction of 1 and/or 3, and let $C_{1,3}$ be the component set of parts present in the direction of 1 and/or 3, i.e.,

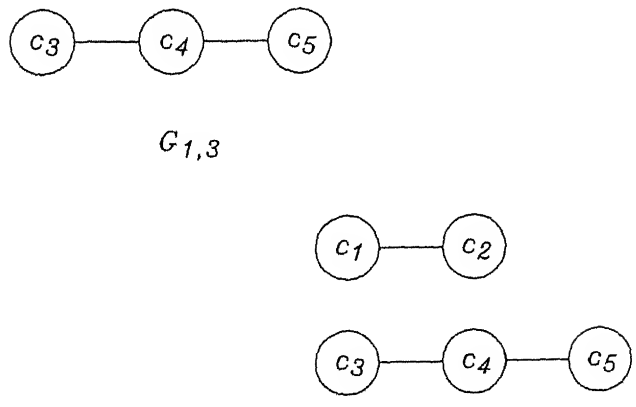
$$C_{1,3} = \{c_k \mid c_k \text{ has components either in the direction of '1' and/or '3', (i.e., } c_k \text{ is in the subgraph } G_{1,3})\}. \tag{5.5}$$

c_k \ $c_{k'}$	1	2	3	4	5
1	–	1	1	1	
2	1	–			1
3	1		–	1	
4	1		1	–	1
5		1		1	–

(a)

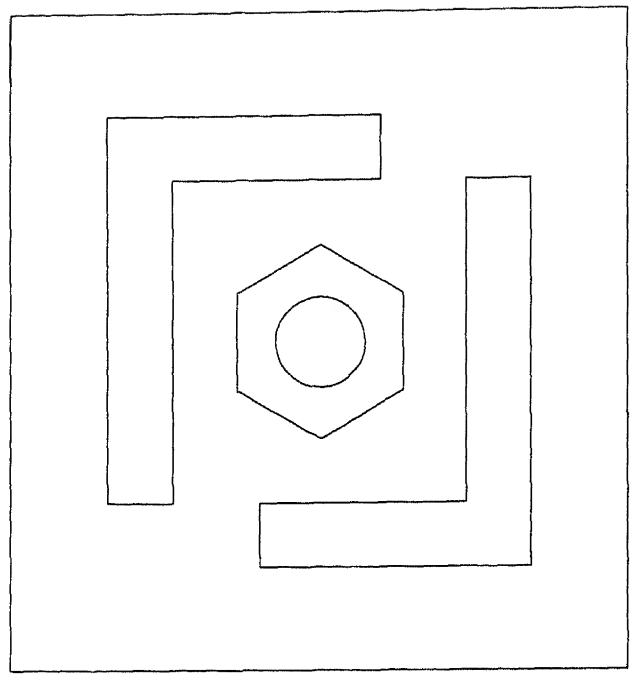
c_k \ $c_{k'}$	1	2	3	4	5
1	–	1 [5]	1 [2]	1 [5]	
2	1 [7]	–			1 [5]
3	1 [4]		–	1 [1]	
4	1 [7]		1 [3]	–	1 [1]
5		1 [7]		1 [3]	–

(b)

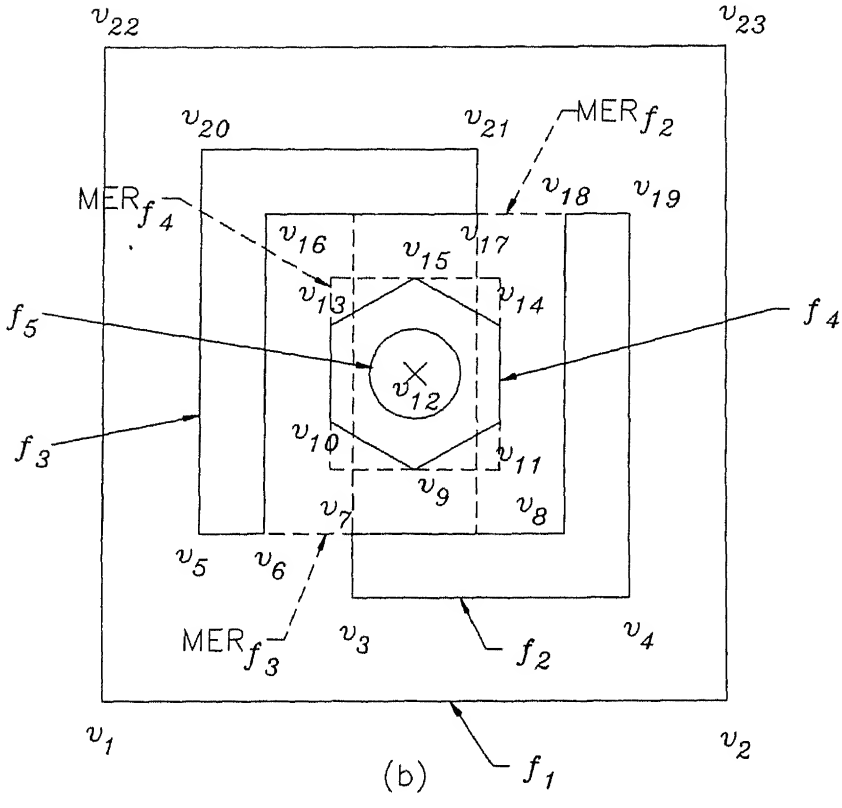


(c)

Figure 5.5. For figure 5 4 : (a) Adjacency component relationship matrix, (b) ACDR table, and (c) Subgraphs of components present in the directions of 1 and 3, 2 and 4, and 1, 3, 5 and 7



(a)



$c_{k'}$ / c_k	1	2	3
1	–	1 [6]	1 [5]
2	1 [8]	–	1 [5]
3	1 [7]	1 [7]	–

(c)

Figure 5.6· Dissimilar components nested layout having intersecting MER of components · (a) Nested layout, (b) Layout graph and MER of components, and (c) ACDR table.

Let RC be the set of components absent in the rows and is given by

$$RC = C \cap C_{1,3}. \tag{5.6}$$

If RC is a null set, then the components are in rows, else they are not in rows Let $G_{2,4}$ represent the subgraph of components having adjacent components present in the direction of 2 or 4, and let CC be the set of components absent in the column Then CC is given by

$$CC = C \cap C_{2,4} \tag{5 7}$$

where $C_{2,4}$ is the component set of parts present in the subgraph ($G_{2,4}$), and is given by

$$C_{2,4} = \{c_k \mid c_k \text{ has component either in the direction of '2' and/or '4' i.e., } c_k \text{ is in the subgraph } G_{2,4}\} \tag{5 8}$$

If CC is a null set, then the components are in columns, else components are not in columns

Thus, if eqn (5 7) and eqn. (5 8) are null sets, then the layout has components in the rows and columns only If they are not null sets, then the layout consists of components in the mid of rows and columns

For example, consider the layout shown in Fig 5.4(a). The components present in the subgraphs $G_{1,3}$ and $G_{2,4}$ shown in Fig 5.5(c) have adjacent components in the directions of 1 and/or 3, and 2 and/or 4 respectively These subgraphs are determined from the ACDR table (Fig. 5 5(b)) of the layout Therefore elements (i.e , components) of $C_{1,3}$ and $C_{2,4}$ are given by

$C_{1,3} = \{c_3, c_4, c_5\}$

and

$C_{2,4} = \{c_1, c_3\}$

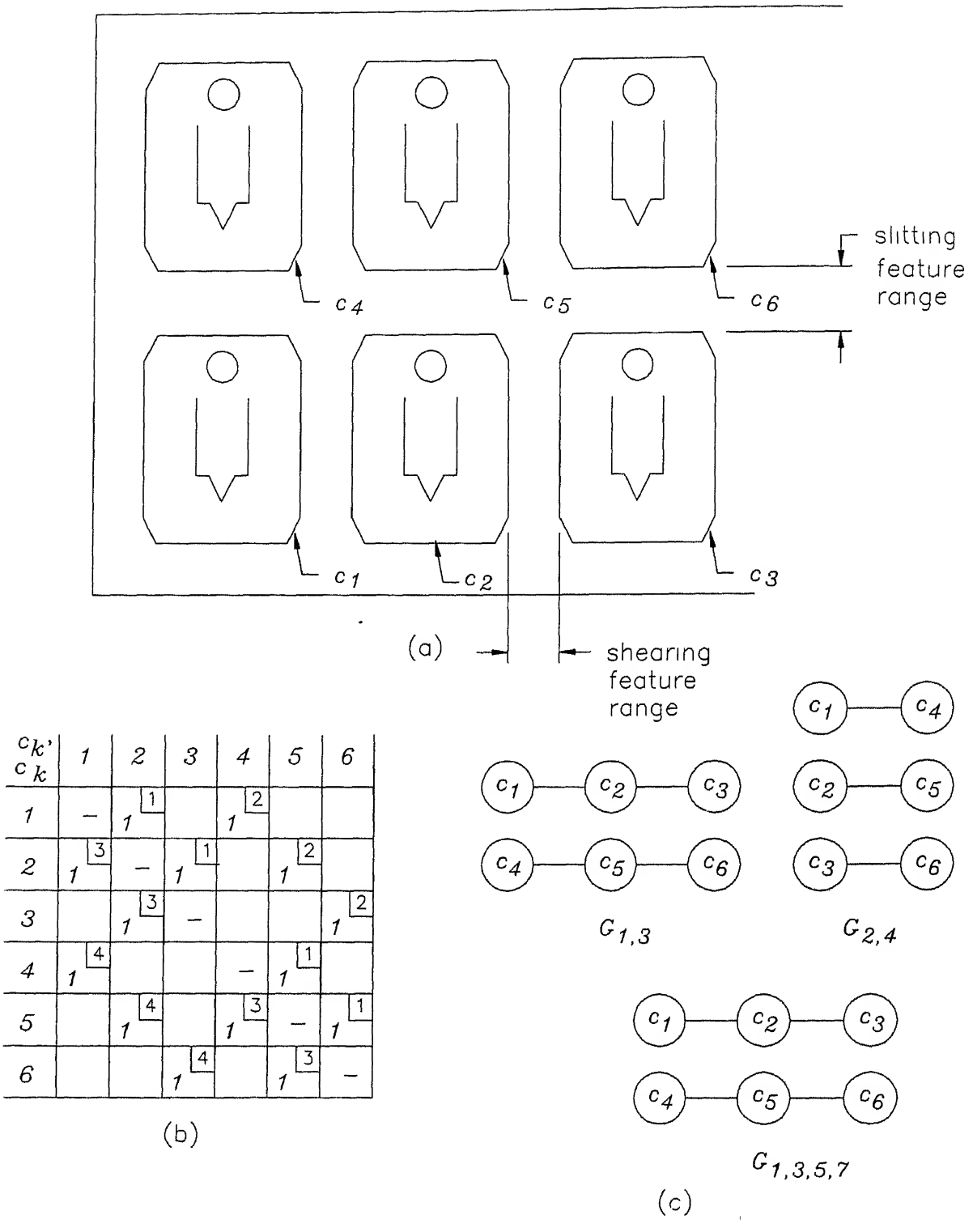
RC and CC are not null sets, and hence the layout is not made up of components only in the rows and/or columns, i.e., it has components present in the mid of rows and columns. It can also be noted that the D_l is not a null set, which indicates that the layout is not for a single component

For the component nested layout shown in Fig. 5 7(a), the ACDR table is given in Fig. 5 7(b). The directional set D_l of layout is determined from the ACDR table, and its elements are $D_l = \{1, 2, 3, 4\}$

and the component set C is given by

$C = \{c_1, c_2, c_3, c_4, c_5, c_6\}$.

Since the D_l set is not a null set, the components present in the subgraphs $G_{1,3}$ and $G_{2,4}$ are determined from the ACDR table. These subgraphs are given in Fig. 5.7(c). The elements of the component sets $C_{1,3}$ and $C_{2,4}$ present are given by



$c_{k'}$ / c_k	1	2	3	4	5	6
1	–	1 [1]		1 [2]		
2	1 [3]	–	1 [1]		1 [2]	
3		1 [3]	–			1 [2]
4	1 [4]			–	1 [1]	
5		1 [4]		1 [3]	–	1 [1]
6			1 [4]		1 [3]	–

Figure 5.7 A similar type of components nested layout (a) Nested layout, (b) ACDR table, and (c) Subgraphs of components present in the directions of 1 and 3, 2 and 4, and 1, 3, 5 and 7.

$C_{1,3} = \{c_1, c_2, c_3, c_4, c_5, c_6\}$
and
$C_{2,4} = \{c_1, c_2, c_3, c_4, c_5, c_6\}$.

The set RC is given by ·

$RC = C \cap C_{1,3} = \emptyset$.

Since RC is a null set, the components in the layout are in rows Similarly the set CC is also found to be a null set.

$CC = C \cap C_{2,4} = \emptyset$.

Therefore, the layout is made up of components present in only rows and columns By vertex fusion methodology (where nodes represent components in the subgraphs), it is determined that the subgraphs $G_{1,3}$ and $G_{2,4}$ (Fig. 5 7(c)) have two and three sub-subgraphs, respectively. Also, components present in each sub-subgraph are determined by vertex fusion method Hence, the nested layout has two rows and three columns Thus the given layout is made up of two rows and three columns, and the direction of layout is '1', since the raw material is of coil type (determined as per the methodology given in the Section 5.3), and the length of coil is in the direction of '1'.

Implementation

During implementation, the ACDR linked list is travelled, and the components present in the required directions are determined. The pseudo code is given below to travel the ACDR linked list and for finding the graph and components present in the direction '1' and '3'.

```
begin
 j = 1
 for (k = 1 to n) {
  for (k' ≠ NULL) {
   if (d_kk' == 1 or d_kk' == 3) {
    if c_k is not present in the set C_1,3 {
     C_1,3 ⇐ c_k
     }
    if (d_kk' == 1 and c_k ∉ G_1,3) {
     G_1,3_j ⇐ c_k
    }
   }
  }
  for all k' linked to k {
   if (there is d == 3 and there is no d == 1 and c_k ∉ G_1,3) {
```

$G_{1,3_j} \Leftarrow c_k$
$j = j + 1$
}
}
}
end.

Similarly, the components present in the set $C_{2,4}$ and graph $G_{2,4}$ are determined

After determination of the direction of layout, type of layout, ACDR table and raw material type, shearing features are recognized as explained in the following section

5.5 Recognition of shearing features

In the present work, shearing features are identified in two steps

In the *first step*, the nested layout graph obtained by preprocessing the geometrical data is used to identify the features These features are classified into various feature sets Raw material entity group and components are identified, and also ACDR table is extracted from the layout graph (these steps have already been discussed in the previous sections)

In the *second step*, shearing features are characterized by developing a set of rules by relating them geometrically and topologically, and using these rules, shearing features are identified (these are discussed below)

Shearing features are recognized based on the classification system of shearing features given in Fig 2.6. Characterization and recognition process are explained in the following sections. In a component nested layout, all features which form potential candidates for manufacturing the component from the given layout are identified, as this helps in the development of alternative process plans

5.5.1 Recognition of inside shearing features

Inside shearing features are characterized by the following parameters.

1. Piercing and lancing features are present inside a component and hence they are called inside features

2 Graph of the feature is of circular (loop) type in case of a piercing feature, and acyclic (path) type in case of a lancing feature.

Thus the elements of inside feature set will be either piercing or lancing features.

For example, in Fig 4.1, elements inside the feature set are given in Table 4.5. In the inside feature set N, features f_4, f_5, f_6, f_7, f_9, f_{10} and f_{11} are recognized as piercing features, and f_8 as lancing feature

5.5.2 Recognition of boundary shearing features

Boundary shearing features are present along the boundary of a component as defined earlier in Chapter 2. The features are blanking, cutoff, notching and parting. Recognition process for these features along with their characteristics is given below :

Recognition of blanking feature

Blanking is a process in which *the complete outline of a component is separated from the parent material in a single press stroke* (Metals Handbook, Vol. 4, 1979). Therefore, the recognition of a blanking feature is done by matching the shape and size of the element of B with the shape and size of the tool stored in the tool database Further, the press on which the part has to be manufactured should be examined to determine whether the selected tool can be mounted on it or not As such, for subgraphs of all the elements belonging to boundary feature set B, *blanking operation is considered as a potential candidate* for manufacturing the component from the layout

It should be noted that during the feature recognition process, if cutoff feature is identified in the component layout then it eliminates the presence of blanking feature from the layout, since the component cannot be manufactured from the layout by the blanking operation.

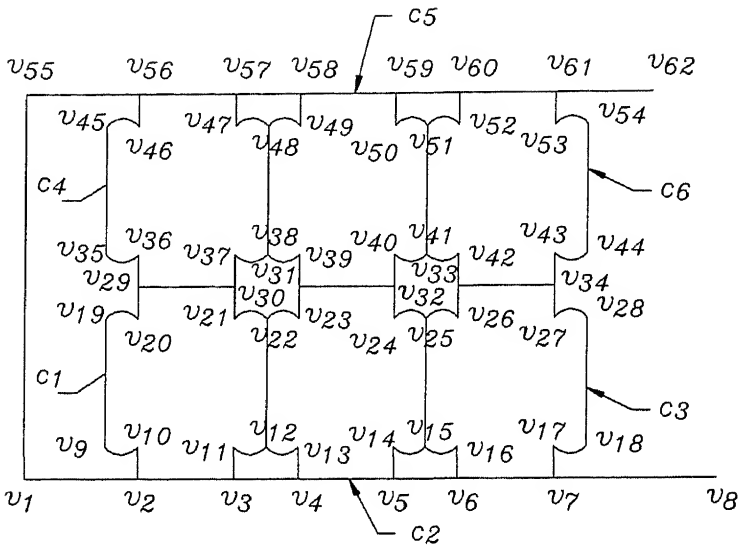
For example, in Fig. 4.1, the boundary features f_2 and f_3 represent the blanking features of the components c_1 and c_2 respectively, as the component can be separated from the parent material along its boundary in a single stroke. Similarly in Fig 5.4, if blanking tools for features f_2, f_3, f_{13}, f_{14} and f_{15} are available, then the components can be separated out from the parent material.

Recognition of cutoff feature

Cutoff is a process which *cuts along a line to produce blanks without generating any scrap during the cutting operation* (Metals Handbook, Vol 4, 1979). A nested layout having cutoff features is shown in Fig. 5.8, and the characteristics of the same are given below :

1. Cutoff feature *may be* present only when there are more than one component in a layout.

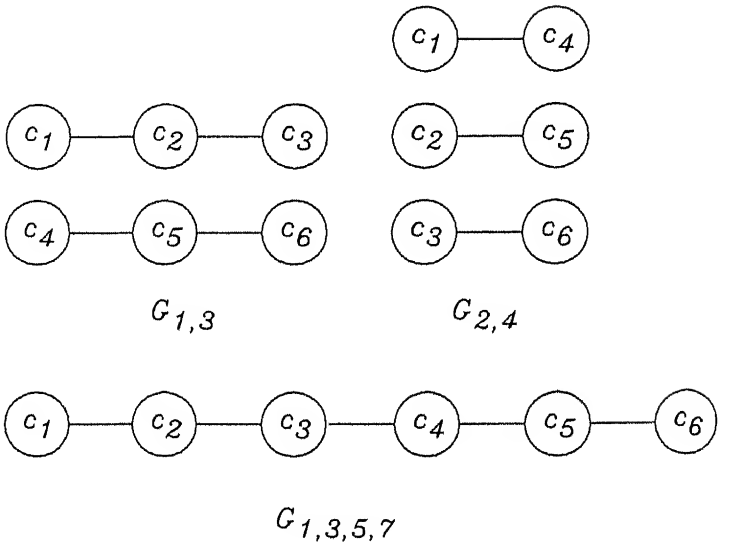
2. A cutoff feature *exists* when an elements of the boundary feature set of the adjacent



(a)

c_k' / c_k	1	2	3	4	5	6
1	–	1 / 1		2 / 1	5 / 1	
2	3 / 1	–	1 / 1	6 / 1	2 / 1	5 / 1
3		3 / 1	–		6 / 1	2 / 1
4	4 / 1	8 / 1		–	1 / 1	
5	7 / 1	4 / 1	8 / 1	3 / 1	–	1 / 1
6		7 / 1	4 / 1		3 / 1	–

(b)



(c)

Figure 5.8: A similar type of components nested layout having cutoff feature · (a) Nested layout, (b) ACDR table, and (c) Subgraphs of components present in the directions of 1 and 3, 2 and 4, and 1, 3, 5 and 7.

components have common vertices or if vertices of the boundary feature of base component or adjacent component fall on the surface of the boundary feature of other component.

During implementation, if a component nested layout is created for the similar type of boundary features, then the search procedure can be reduced in finding the common vertices present between the adjacent components. Also, vertices of the base component falling on the surface of the adjacent component and vice-versa. Various situations, in which a vertex of the base component falls on the surface of the adjacent component are shown in Fig. 5.9, and they depend on the adjacency direction of the components Information about similar components nested layout is asked interactively from the designer to minimize the search procedure for identifying the vertices belonging to cutoff feature Even in the absence of this information (similar and dissimilar components), the system works well, but takes a little more time to identify the cutoff feature

The following mathematical model has been developed for identifying a cutoff feature.

In the ACDR table, the element of the cells are either '0' or '1'. If the element is '1', the components are adjacent to each other, otherwise not. For adjacent components, the subgraphs of elements of B (b_j and $b_{j'}$) are travelled in the clockwise direction to determine the common vertices between them. The relationship is given by

$$v_{i,j} = v_{i,j'}; \quad v_{i,j} \in V_j,\ v_{i,j'} \in V_{j'},\ f_j \in B,\ f_{j'} \in B \text{ and } f_{j'} \neq f_j. \tag{5.9}$$

In case, the base vertex (the vertex of the base component) is not common for the subgraphs b_j and $b_{j'}$, it is tested for determining whether it falls on the surface of the adjacent component or not. If the base vertex falls on an adjacent component surface, then it has to be collinear with respect to vertices of edges belonging to the surface of the adjacent component under consideration . The collinearity of vertices is determined from the following determinant, whose value will be zero for collinear points.

$$\begin{vmatrix} v^x_{i,j'} & v^y_{i,j'} & 1 \\ v^x_{i,j} & v^y_{i,j} & 1 \\ v^x_{i+1,j'} & v^y_{i+1,j'} & 1 \end{vmatrix}$$

$$v_{i,j} \in V_j,\ v_{i,j'} \in V_{j'},\ f_j \in B,\ f_{j'} \in B, \text{ and } f_{j'} \neq f_j. \tag{5.10}$$

Further, if the vertex is collinear then the containment of the vertex within the surface is determined by the rules given in Table 5.8. Thus, cutoff feature between the two adjacent components is determined and if it exists, then blanking feature will not be present between the adjacent components under consideration.

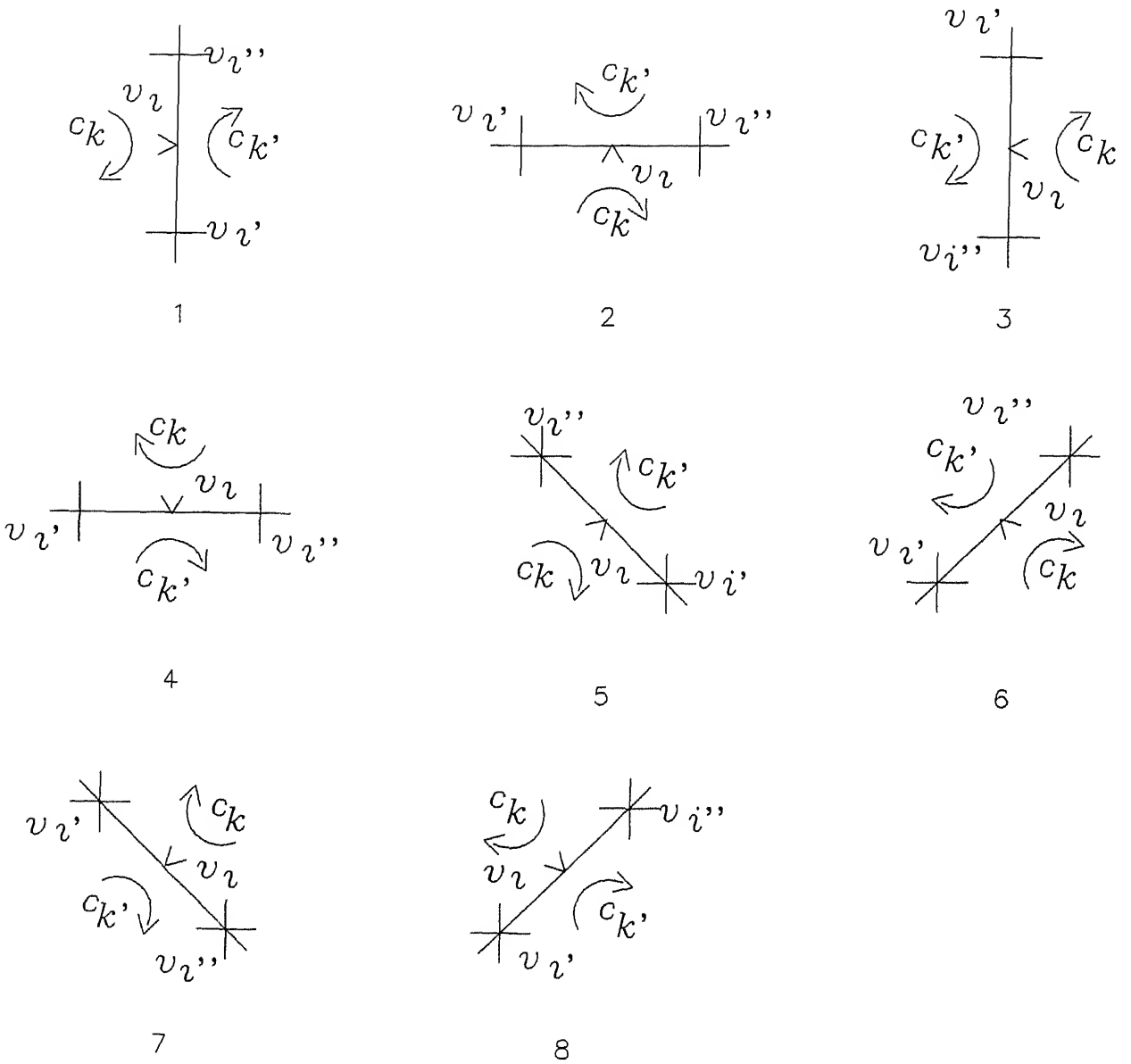


Figure 5.9: Modes of a vertex of the base component falling on the adjacent component surface (for similar type of components).

Directions	*Rules*
1	$v^y_{i'j'} \leq v^y_{ij} \leq v^y_{i''j'}$
2	$v^x_{i'j'} \geq v^x_{ij} \geq v^x_{i''j'}$
3	$v^y_{i'j'} \geq v^y_{ij} \geq v^y_{i''j'}$
4	$v^x_{i'j'} \leq v^x_{ij} \leq v^x_{i''j'}$
5	$v^y_{i'j'} \leq v^y_{ij} \leq v^y_{i''j'}$ and $v^x_{i'j'} \geq v^x_{ij} \leq v^x_{i''j'}$
6	$v^y_{i'j'} \leq v^y_{ij} \leq v^y_{i''j'}$ and $v^x_{i'j'} \leq v^x_{ij} \leq v^x_{i''j'}$
7	$v^y_{i'j'} \geq v^y_{ij} \geq v^y_{i''j'}$ and $v^x_{i'j'} \leq v^x_{ij} \leq v^x_{i''j'}$
8	$v^y_{i'j'} \leq v^y_{ij} \leq v^y_{i''j'}$ and $v^x_{i'j'} \leq v^x_{ij} \leq v^x_{i''j'}$
	$\forall \; v_i \in V_j;\ v_{i'} \in V_{j'}; v_{i''} \in V_{j'};\ i' \neq i''$, $j' \neq j$, $f_j \in B$ and $f'_j \in B$

Table 5.8: Rules for determining whether the base vertex falls within the segment of the surface of the adjacent component under consideration or not

cutoff features (cf) :
units . mm

No.	Feature No.	Vertices	
1	$cf_{1,2}$	v_{12}	v_{22}
2	$cf_{1,4}$	v_{29}	v_{30}
3	$cf_{2,3}$	v_{15}	v_{26}
4	$cf_{2,5}$	v_{31}	v_{32}
5	$cf_{3,6}$	v_{33}	v_{34}

Table 5.9. Cutoff features present in the component layout given in Fig 5 8

Implementation

The above methodology is implemented for both similar and dissimilar type of boundary features, and the flowchart is given in Fig 5 10. The output of the system for the component layout shown in Fig 5 8 is given in Table 5.9

Recognition of notching feature

Notching operation is a process in which *an individual punch removes a piece of metal from the edge of the blank or strip* One of the reasons for notching operation is *to cut a part of the outline of the blank that would be difficult to cut otherwise* (Metals Handbook, Vol. 4, 1979). Therefore a notching feature is present when the shape and size of the complete boundary of a component cannot be produced by blanking operation. The characteristics of the notching features are :

1. Notching feature is present along the boundary feature of the component.

2 Feature is present, when the shape of the boundary feature is not rectangular, and its sides are not parallel to the X and Y axes.

To identify a notching feature, initially minimum raw material required to manufacture the component is determined. It is assumed that the minimum raw material required for a component under consideration is the minimum enclosing rectangle (MER) for the component, with sides parallel to the X and Y axes. In Fig. 5.11(a), the dashed enclosure

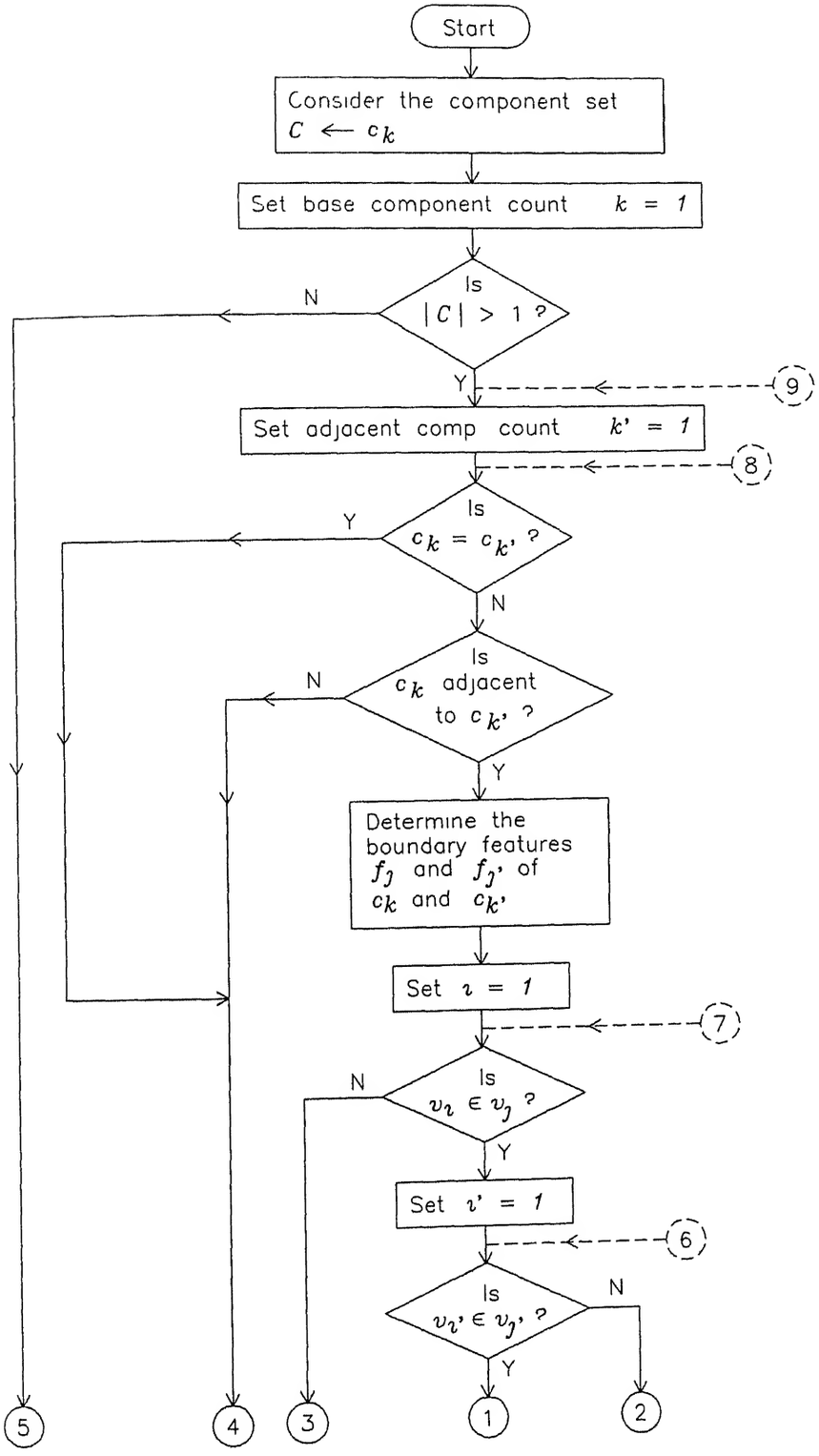


Figure 5.10: Flowchart for recognizing cutoff feature.

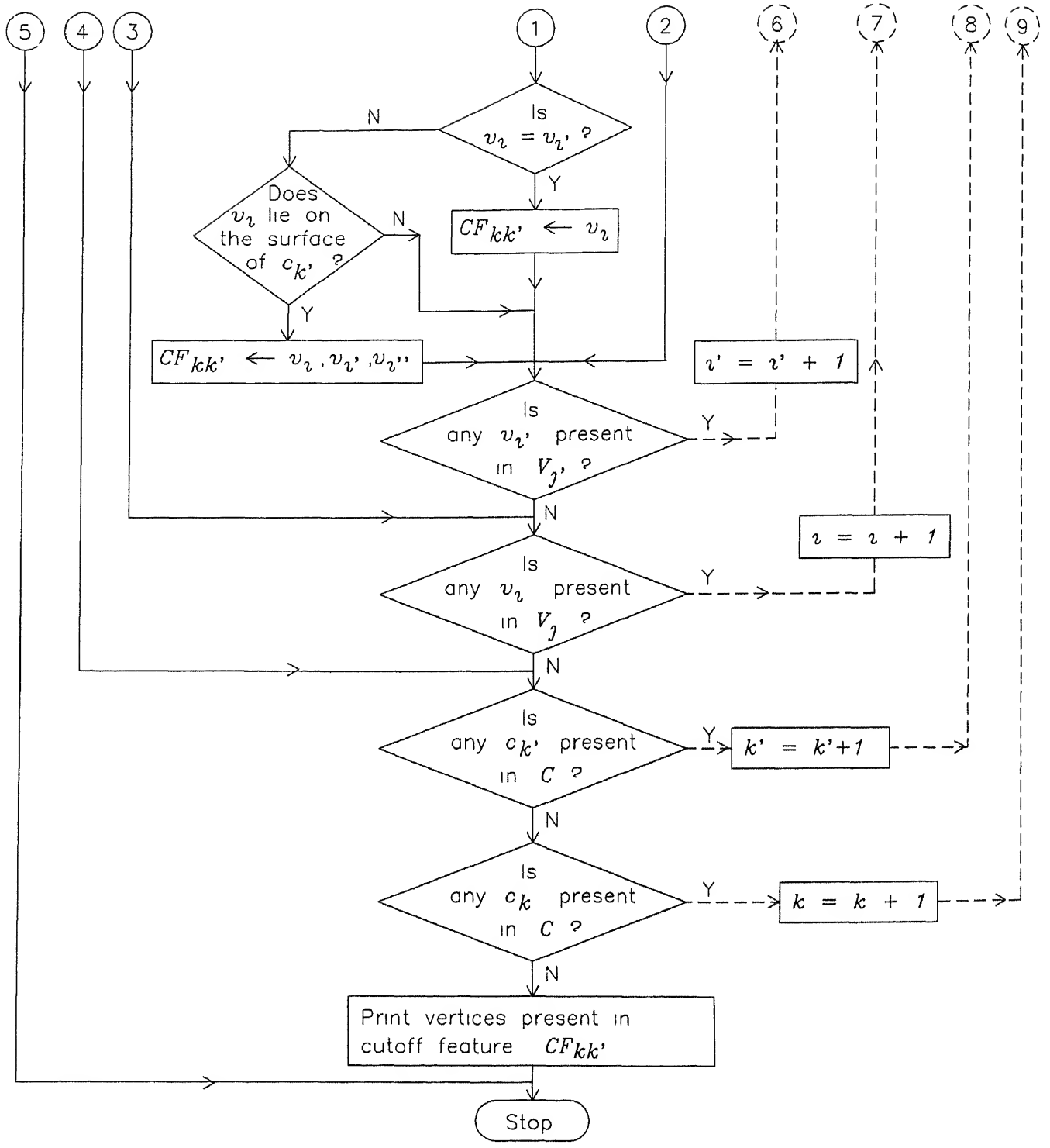


Figure 5.10: Flowchart for recognizing cutoff feature (continued).

surrounding the component represents the MER shaped raw material required for manufacturing the given component. Vertices and edges of MER of a feature are determined as explained in the Section 4 3

In the second step, convexity or concavity (Roger and David, 1985) is determined for every vertex of the element belonging to the boundary feature set of the component The convexity or concavity is obtained by the vector cross product. Vector cross product for the vertex under consideration is obtained by the cross product of the vectors with respect to previous vertex and next vertex respectively. The vector cross product is positive in $\vec{k}$ direction for a convex vertex and negative in $\vec{k}$ direction for a concave vertex. The vector cross product is determined as given below ·

$$\begin{aligned} \vec{V}_{i-1} &= (v_{i-1}^x - v_i^x)\ \vec{i} + (v_{i-1}^y - v_i^y)\ \vec{j} \\ \vec{V}_{i+1} &= (v_i^x - v_{i+1}^x)\ \vec{i} + (v_i^y - v_{i+1}^y)\ \vec{j} \\ \vec{V}_i &= \vec{V}_{i-1} \times \vec{V}_{i+1} \\ & \qquad \forall \quad v_i \in V \end{aligned} \tag{5.11}$$

$$\text{If } \vec{V}_i = \begin{cases} +ve \text{ in } \vec{k} \text{ direction}, & \text{vertex } v_i \text{ is of convex type} \\ -ve \text{ in } \vec{k} \text{ direction}, & \text{vertex } v_i \text{ is of concave type} \end{cases} \tag{5.12}$$

If a vertex has a negative bulge (convex arc) or a positive bulge (concave arc) as an attribute in the polyline, then the convexity or concavity of the vertex is determined as follows :

(i) In case of a negative bulge, pseudo vertices (p_1 and p_2 in Fig. 5.11(b)) which are intersection of lines parallel to X axis and Y axis passing through the center of the arc under consideration and the arc itself, are determined. These pseudo vertices are inserted in the circular subgraphs as nodes and they are connected by edges which represent lines, such that they form a part of the loop. Vector cross product is determined for the vertices and for the pseudo vertices. The convexity or concavity of the vertices and pseudo vertices is determined using eqn 5.12.

(ii) In case of a positive bulge, the start and end vertices of an arc are always of convex type.

In Fig. 5.11(b), positive and negative signs represent convexity and concavity respectively. Once minimum raw material required for the component and convexity or concavity of the vertices of elements belonging to the boundary feature set are determined, the component is tested for the presence of notching features.

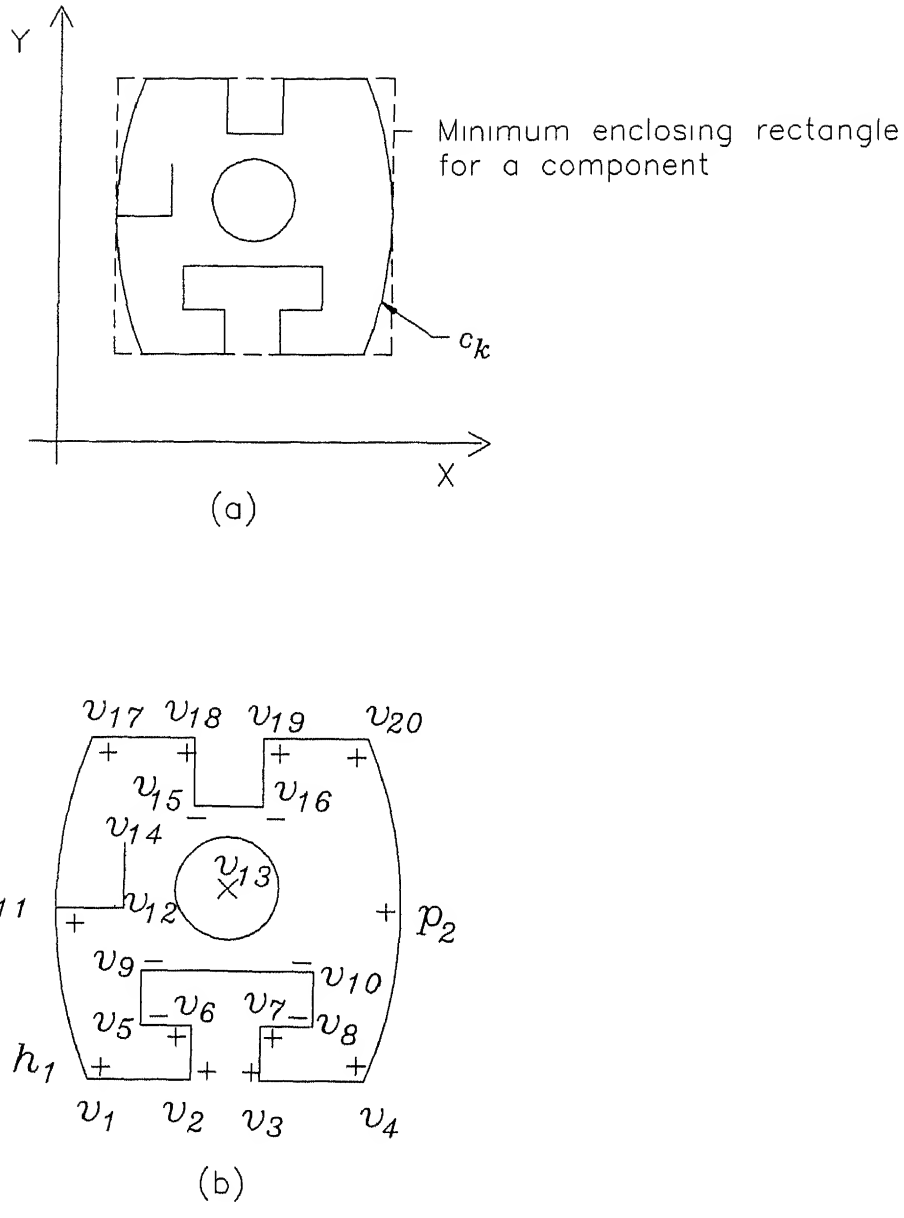


Figure 5 11: (a) Minimum enclosing rectangle for a component, and (b) Representation of vertices and pseudo vertices, along with their convexity and concavity

Let P_j (includes vertices of j^{th} boundary feature b_j, pseudo vertices and vertices of MER of b_j) represent the vertex set of duplet vertices of b_j belonging to B i.e ,

$$P_j = V'_j \cup V_{MER_j}$$

$$\forall \quad f_j \in B \tag{5.13}$$

where V_{MER_j} is the vertex set of MER of j^{th} boundary feature

A notching feature is present if one of the following conditions is satisfied

$$\vec{V}_{ij} \text{ is in the negative direction of } \vec{k}, \quad \forall \ i \in I_j \tag{5.14(a)}$$

$$(v^x_{min,j}, v^y_{min,j}) \notin P_j \tag{5.14(b)}$$

$$(v^x_{min,j}, v^y_{max,j}) \notin P_j \tag{5.14(c)}$$

$$(v^x_{max,j}, v^y_{max,j}) \notin P_j \tag{5.14(d)}$$

$$(v^x_{max,j}, v^y_{min,j}) \notin P_j \tag{5.14(e)}$$

$$\forall \quad f_j \in B.$$

Notching features are recognized by travelling a subgraph in the clockwise direction from the head vertex of a component If eqn. (5.14(a)) is satisfied, the presence of a notching feature is confirmed, and the feature is identified by the following procedure.

When a vertex is of concave type (v_{15} in Fig 5.11(b)), the subgraph is travelled in the backward direction till a convex vertex (v_{18}) is met, such that it lies on the boundary of the minimum raw material (MER) required by the component. Now, from the concave vertex (v_{15}), the subgraph is travelled in the forward direction till a convex vertex (v_{19}) lying on the boundary of the MER of the component is reached Notching feature is formed by the vertices present between these two convex vertices (i.e, v_{18}, and v_{19}).

In case any one of the equations eqns. (5.14(b) to 5.14(e)) is satisfied, then the procedure for identifying the notching feature is as follows ·

As the subgraph is travelled in clockwise direction (Fig. 5 11(b)), the last vertex (v_1) lying on the first edge and first vertex (p_1) of the second edge is recognized as the notching feature (i.e., v_1, p_1, and the one of the vertices of MER in Fig. 5.11(b)).

A part after notching operation can be blanked using blanking tool having MER shape and size MER is one of the shapes of blanking feature that can be used to get the part after notching the notching features. Total number of shapes of a blanking feature that can be obtained with different combination of notching features is given by

$$1 + \frac{nf * (nf + 1)}{2} \tag{5 14}$$

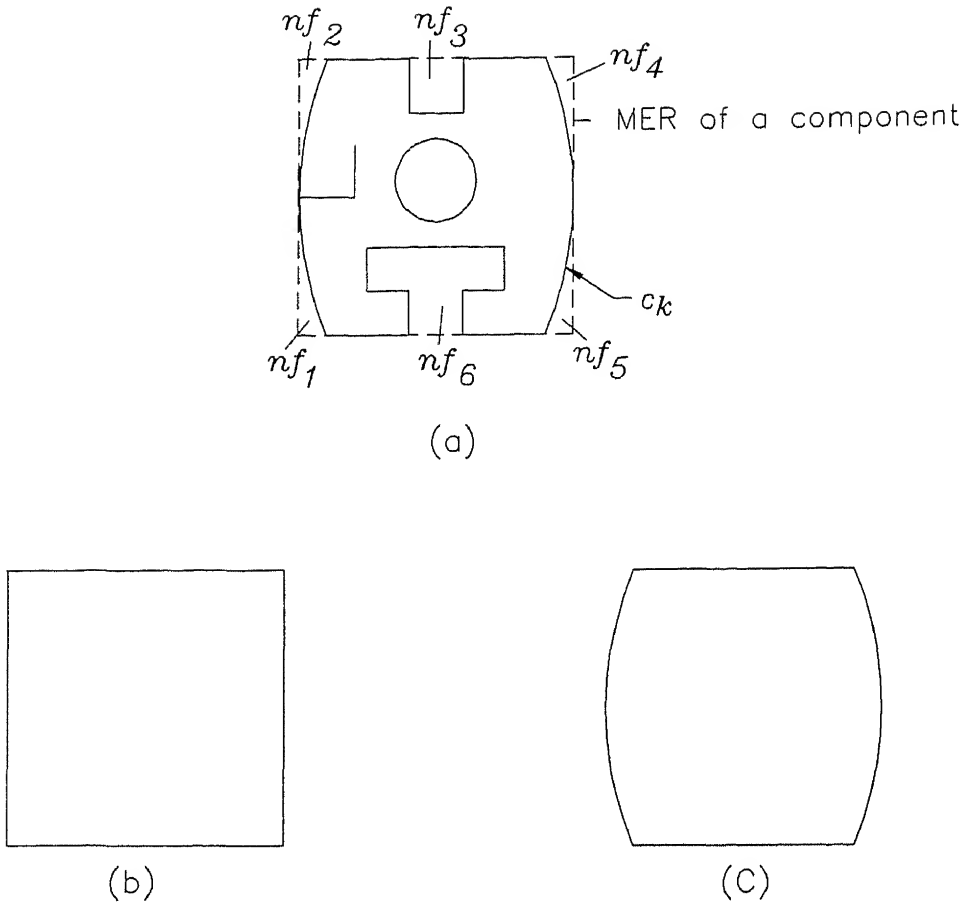


Figure 5.12· Various tools that can be used for blanking a component .(a) Representing MER of a component and notching features, (b) Blanking tool having same shape and size as that of the MER, and (c) Another shape of the blanking tool.

where nf represents the number of notching features present w.r.t. MER of a part. It is to be noted that the notching features are not divided further. Fig. 5.12(b) shows MER shape and size of the blanking feature. The part can be blanked after notching the notching features nf_1, ., nf_6 shown in Fig. 5.12(a). One of the shapes of blanking feature obtained with different combinations of notching features is shown in Fig. 5.12(c). This shape and size is obtained by subtracting the shapes of notching features (nf_1, nf_2, nf_4, and nf_5) from MER shape. The part can be blanked after notching nf_3, and nf_6, by using the shape and size of the blanking feature shown in Fig. 5.12(c).

Implementation

The flowchart for finding notching features from a layout is given in Fig. 5.13. The input to this module is information about MER of the components and ACDR table of the layout. For the component layout shown in Fig. 5.2, the output is given in Table 5.10.

notching features (nf) :
units : mm
component : c_1
feature : f_2

Sl. No.		x	y	rad /bul	start ang	end ang
1	$nf_{1,1}$:					
	position .	57 500	214 867			
		57 500	244.867	30 000	270.000	234.315
		40 000	220.500			
		40 000	214.867			
2	$nf_{2,1}$:					
	position :	57.500	241.183			
		57.500	214 183	30.000	125.685	90.000
		40 000	235 500			
		40.000	241.183			
3	$nf_{3,1}$:					
	position :	57.500	241 183			
		57.500	214 183	30.000	90.000	54.315
		75.000	235 500			
		75.000	241.183			
4	$nf_{4,1}$:					
	position :	75.000	220.500			
		57.500	244.867	30.000	305.685	270.000
		57.500	214.867			
		75 000	220 500			

Table 5.10: Notching features present in the component layout given in Fig 5 2

Recognition of parting feature

Parting is a process which *separates blanks from parent material by cutting away a strip of material between them* (Metals Handbook, Vol. 4, 1979). Parting feature is characterized as given below :

1. Parting feature is present between two adjacent components.

2. Parting feature is present only in coil or strip type of raw material.

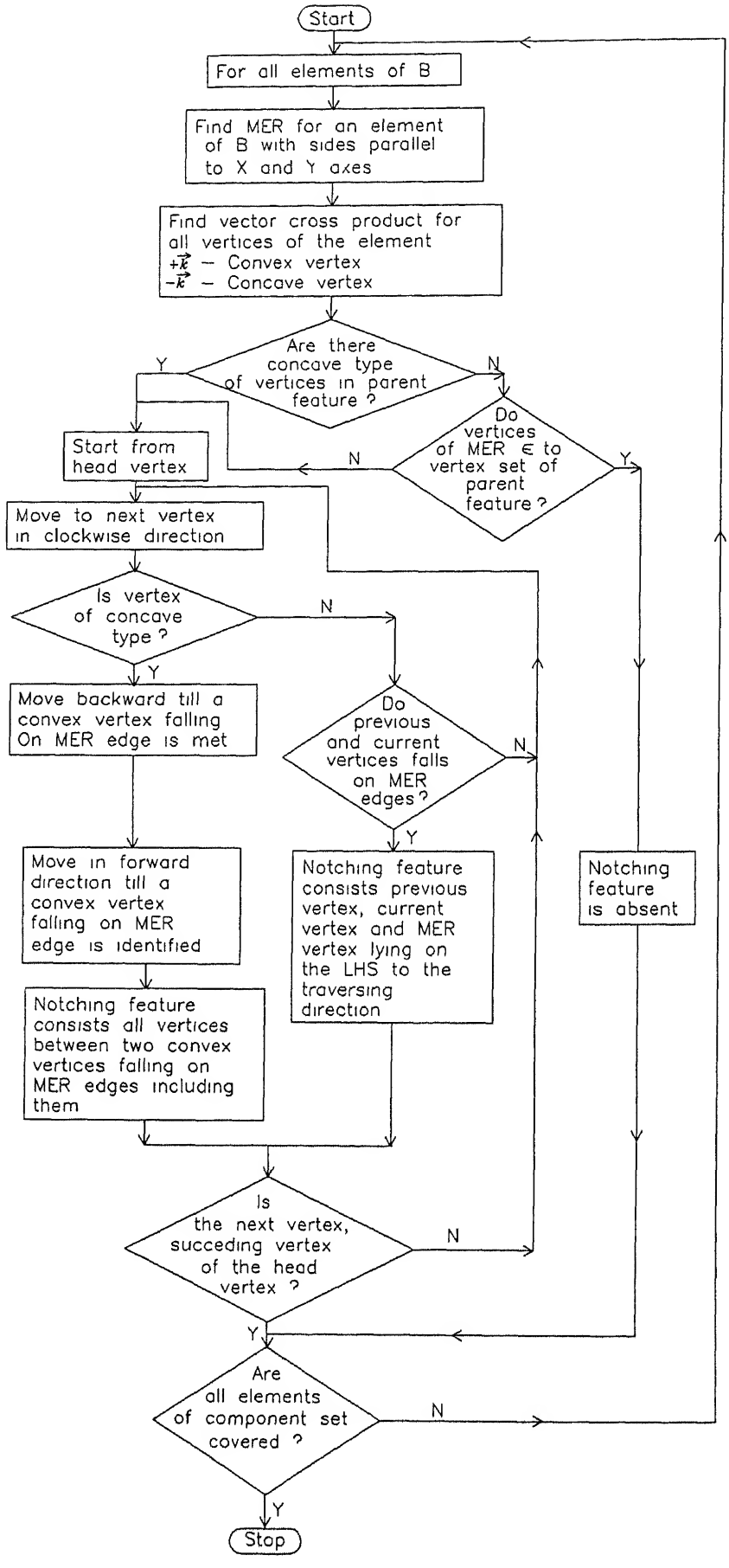


Figure 5.13: Flowchart for recognizing notching features.

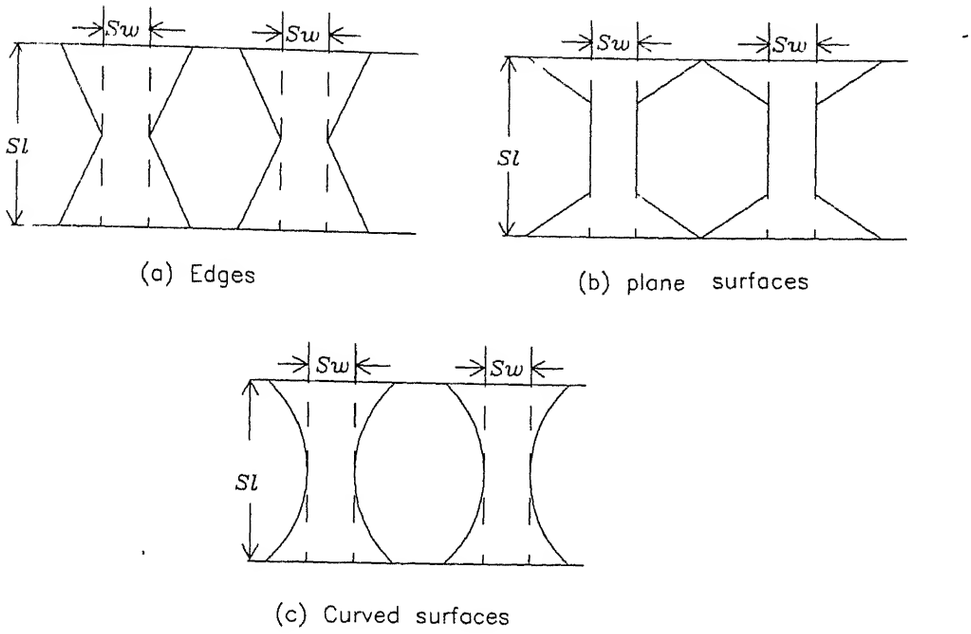


Figure 5.14: Parting features for a layout in the direction of X axis

3. Layout should contain either single row or single column of components.

4. The slug between the two adjacent components forms the parting feature

As in the case of blanking, parting operation is possible only if the shape and size of the tool is similar to that of the slug, and the slug is removed by one or utmost two blows.

The raw material type, component nested layout type and ACDR table determined in the previous sections are the input to this module

Parting features are always perpendicular to the direction of the component layout The slug is assumed to be rectangular in shape and various cases are shown in Fig. 5.14 The length and width of the slug (parting feature) are determined for the component layout in the direction '1'.

$Sw_{kk'}$ = slug width between k^{th} and k'^{th} component.
Sl = slug length.
st = subscript for strip type of raw material.
co = subscript for coil type of raw material

$$Sw_{kk'} = (len^{x}_{max,k}) - (len^{x}_{min,k'}) \quad \exists\ a_{kk'} = 1, \quad \forall\ k \in K,\ k' \in K \text{ and } k \neq k' \tag{5.15}$$

$$Sl_{st} \text{ or } Sl_{co} = (v^{y}_{max,k} - v^{y}_{min,k}) \quad \forall\ k \in K. \tag{5.16}$$

In eqn 5.16, Sl_{st} or Sl_{co} is slug length, and it is dependent on the type of raw material. It can be noted that the length (i e., slug length) of the parting feature remains constant for all the components

Similarly, length and width of parting features are determined for a layout in the direction '2' It should be noted that the parting feature is absent in the layout, if cutoff feature is present between the adjacent components.

Implementation

Initially the type of layout and the raw material type are identified. If conditions '2' and '3' given in characterization of parting features are satisfied, then only parting feature will be identified from the nested layout.

Algorithm

Step 1. Length of the slug is equal to the width of the raw material.

Step 2. Width of the slug is given by minimum rectangle present between the adjacent components

Step 3. Size of the parting feature is equal to the length and width of the slug identified

Step 4. Goto Step 1 until $(nc-1)^{th}$ component in the component set is considered as a base component, where nc is the number of components present in the layout

Step 5. Print the coordinate values of the parting features, and stop the process.

For the Fig 5 2, it has been determined that the components are in a single row. Thus, after checking the existence of parting features, they are recognized by the above method, and the output is given in Table 5 11.

5.5.3 Recognition of outside features

Outside operations viz., *shearing* and *slitting* are dependent upon the type of raw material, nesting of components, and the layout direction. The outside features are recognized as follows .

Recognition of Slitting feature

Slitting is a process in which *the coil or strip is cut into narrower coils or strips* (Metals Handbook, Vol. 4, 1979). Slitting feature has been characterized as given below :

1. Slitting feature is present in coil or strip type of raw material.

2 Slitting feature is present along the direction of the layout.

The following methodology is developed to identify the presence of the slitting feature in the direction '1'.

parting features (pf)
units : mm

Sl No		x	y
1	$pf_{1,2}$: position .	75 000	205 000
		87.500	205.000
		75.000	251 000
		87.500	251 000
2	$pf_{2,3}$. position	122.500	205.000
		135.000	205 000
		122.500	251 000
		135 000	251 000

Table 5.11· Parting features present in the component layout given in Fig. 5.2

From the ACDR table, subgraphs of components present in the directions 1 and 3, and 1, 3, 5 and 7 are determined, such that they satisfy the following conditions :

1. v^y_{max} of c_k is greater than v^y_{min} of $c_{k'}$ in direction '5', and

2 v^y_{min} of c_k is lesser than v^y_{max} of $c_{k'}$ in direction '7'

Subgraphs of the components present in the above directions for the layout (Fig. 5.4) are given in Fig. 5.5(c). Similarly, for the component layout shown in Fig. 5.7(a), its ACDR table is shown in Fig. 5.7(b), and subgraphs of the components present in the above directions are given in Fig. 5.7(c) Also, in Fig. 5.7(c), subgraphs of the components present in direction 2 and 4 are given.

A slitting feature may be present in the layout, if the component set $C_{1,3,5,7}$ present in the subgraph $G_{1,3,5,7}$ is not a null set, i.e.,

$$\text{If } C_{1,3,5,7} \text{ is } \begin{cases} \emptyset & \text{slitting feature is absent} \\ \text{otherwise} & \text{slitting feature may be present} \end{cases} \tag{5 17}$$

A slitting feature is absent, if the width of the raw material is equal to the length/width of the component or if the nested layout has only one row of components.

In Fig. 5.7(c), $G_{1,3,5,7}$ is not a null graph, and therefore $C_{1,3,5,7}$ is not a null set. Hence, the layout is further processed for recognizing slitting features.

The graph $G_{1,3}$ and $G_{1,3,5,7}$ may be a single graph or disjoint subgraphs If the subgraphs of both $G_{1,3}$ and $G_{1,3,5,7}$ are *single subgraph*, then sets $Z1$ and $Z2$ is determined as given below

$$Z1 = C \cap C_{1,3} \tag{5 18(a)}$$

$$Z2 = C_{1,3} \cap C_{1,3,5,7} \tag{5 18(b)}$$

and the presence of slitting feature in the layout is dependent on the type of set $Z1$ and $Z2$ as stated below

$$\text{If } Z1 \text{ and } Z2 \text{ is } \begin{cases} \emptyset & \text{slitting feature is } \textit{absent} \\ \text{otherwise} & \text{slitting feature is } \textit{present.} \end{cases} \tag{5 19}$$

It should be noted that for *a single row of components* in coil type of raw material, graphs $G_{1,3}$ and $G_{1,3,5,7}$ are single subgraphs, and the set $Z1$ and $Z2$ are *null sets* Therefore, slitting feature will be absent in a given layout.

Once the presence of a slitting feature is identified, its range is determined in the layout by the following procedure.

For each row present in $G_{(1,3,5,7),row}$ the first component $(c_{f,row})$ and last component $(c_{l,row})$ are determined, and are given by :

$$c_{f,row} = \min_{k \in K} c_k \qquad \forall \quad c_k \in G_{(1,3,5,7),row} \tag{5.20(a)}$$

$$c_{l,row} = \max_{k \in K} c_k \qquad \forall \quad c_k \in G_{(1,3,5,7),row}. \tag{5.20(b)}$$

In the above equations, $G_{(1,3,5,7),row}$ is the row^{th} row of subgraph belonging to the $G_{1,3,5,7}$. From these components present between the first and last component in a row, minimum and maximum y coordinate values for these components (i.e , $(v^y_{min,k})$ and $(v^y_{max,k})$ respectively) are determined from the pseudo vertex set V'_j.

$$v^y_{min,k} = \min_{i \in I_j} v^y_{i,j} \qquad | \ f_j \in B, \text{ and } c_k \in G_{(1,3,5,7),\,row} \tag{5.21(a)}$$

$$v^y_{max,k} = \max_{i \in I_j} v^y_{i,j} \qquad | \ f_j \in B, \text{ and } c_k \in G_{(1,3,5,7),\,row} \tag{5.21(b)}$$

Subsequently, minimum and maximum y coordinate values for each row (i.e., $v^y_{min,row}$ and $v^y_{max,row}$ respectively) present in the graph $G_{1,3,5,7}$ are determined.

$$v^y_{min,row} = \min_{c_k \in G_{(1,3,5,7),row}} v^y_{min,k} \qquad \forall \quad row \in G_{(1,3,5,7)} \tag{5.22(a)}$$

$$v^y_{max,row} = \min_{c_k \in G_{(1,3,5,7),row}} v^y_{max,k} \qquad \forall \quad row \in G_{(1,3,5,7)}. \tag{5.22(b)}$$

Slitting feature is present between two rows. The slitting range at the lower side of the row (slr_{lw}) is given by the difference between the minimum y coordinate value of the

slitting features (slf) :
units . mm

Sl No		x	y
1	sf of 1st row .		
	direction .	X axis	
	slr_{up} .	12 500	
	position :	41.000	224 000
		41 000	236 500

Table 5.12 Slitting features present in the component layout given in Fig 5 7

row under consideration ($v^y_{min,row}$) and the maximum y coordinate value of the previous row ($v^y_{max,row-1}$) Similarly, slitting range at the upper side of the row (slr_{up}) is given by the difference between the minimum y coordinate value of the next row ($v^y_{min,row+1}$) and maximum y coordinate value of the row under consideration ($v^y_{max,row}$), i e ,

$$slr_{lw} = v^y_{min,row} - v^y_{max,row-1} \qquad (5\ 23(a))$$

$$slr_{up} = v^y_{min,row+1} - v^y_{max,row} \qquad (5\ 23(b))$$

Thus, the slitting range is determined between rows It should be noted that, if there are more than one row in a layout, then the slr_{up} of the row under consideration and the next row's slr_{lw} will be same. Similarly, for component layout in direction '2', the presence of slitting features and their ranges are determined.

Implementation

In implementation, the row numbers are determined as per the identification process, and they will be in lexicographical order. The flowchart for finding slitting range is given in Fig. 5.15. The output for Figs. 5.4 and 5 7 is given in Tables 5.12, and 5 13, respectively.

Recognition of Shearing feature

Shearing is a process in which *the material is cut to get specified length and width of raw material* (Metals Handbook, Vol 4, 1979). Therefore, shearing features for sheet type of raw material will be parallel to X or Y axis and for coil and strip type of raw material it will be perpendicular to the direction of the layout Shearing feature recognition process is similar to the recognition process for slitting feature.

In case of sheet type of raw material the subgraphs $G_{1,3,5,7}$ and $G_{2,4,6,8}$ are obtained

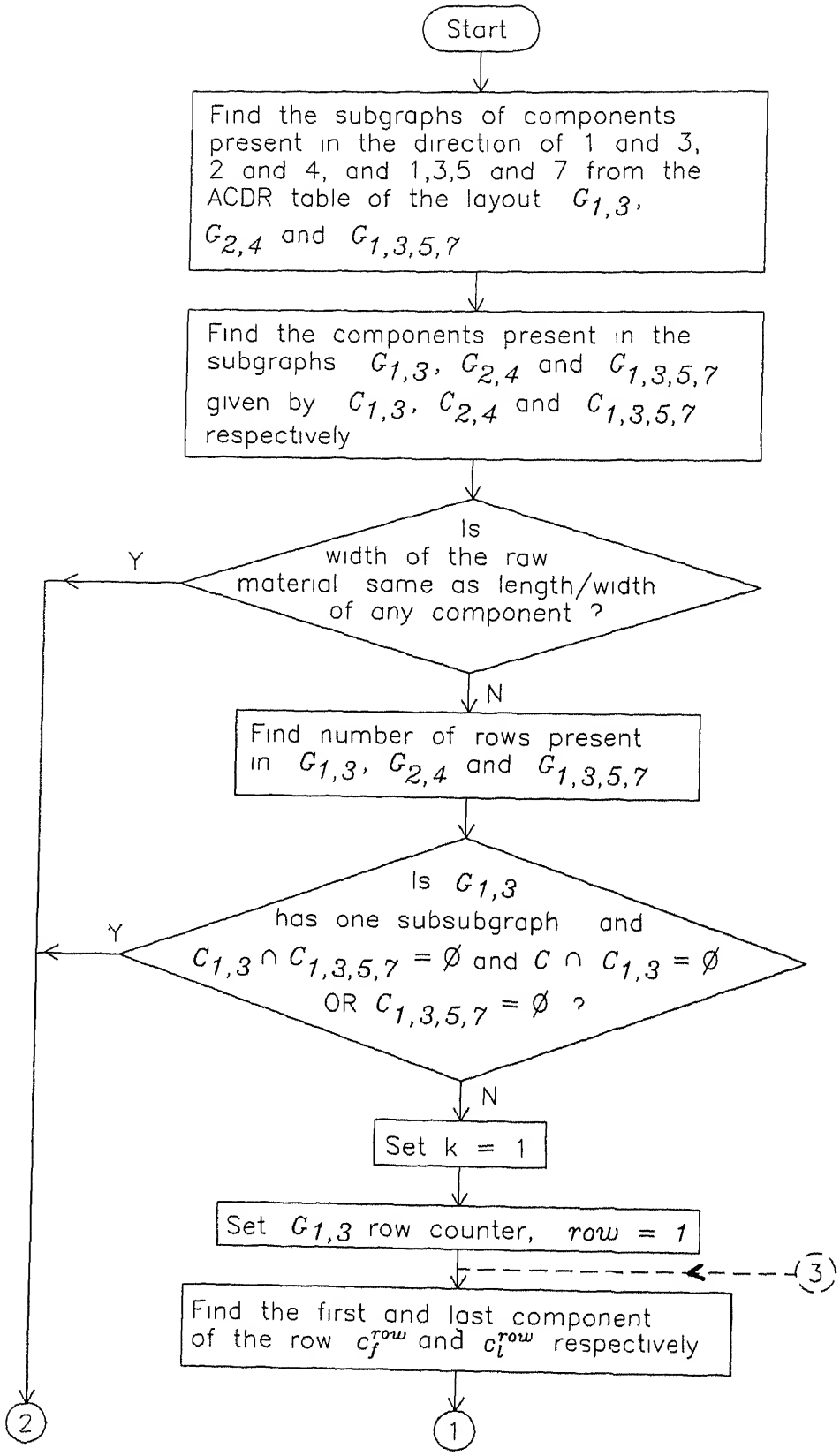


Figure 5.15: **Flowchart for recognizing slitting features.**

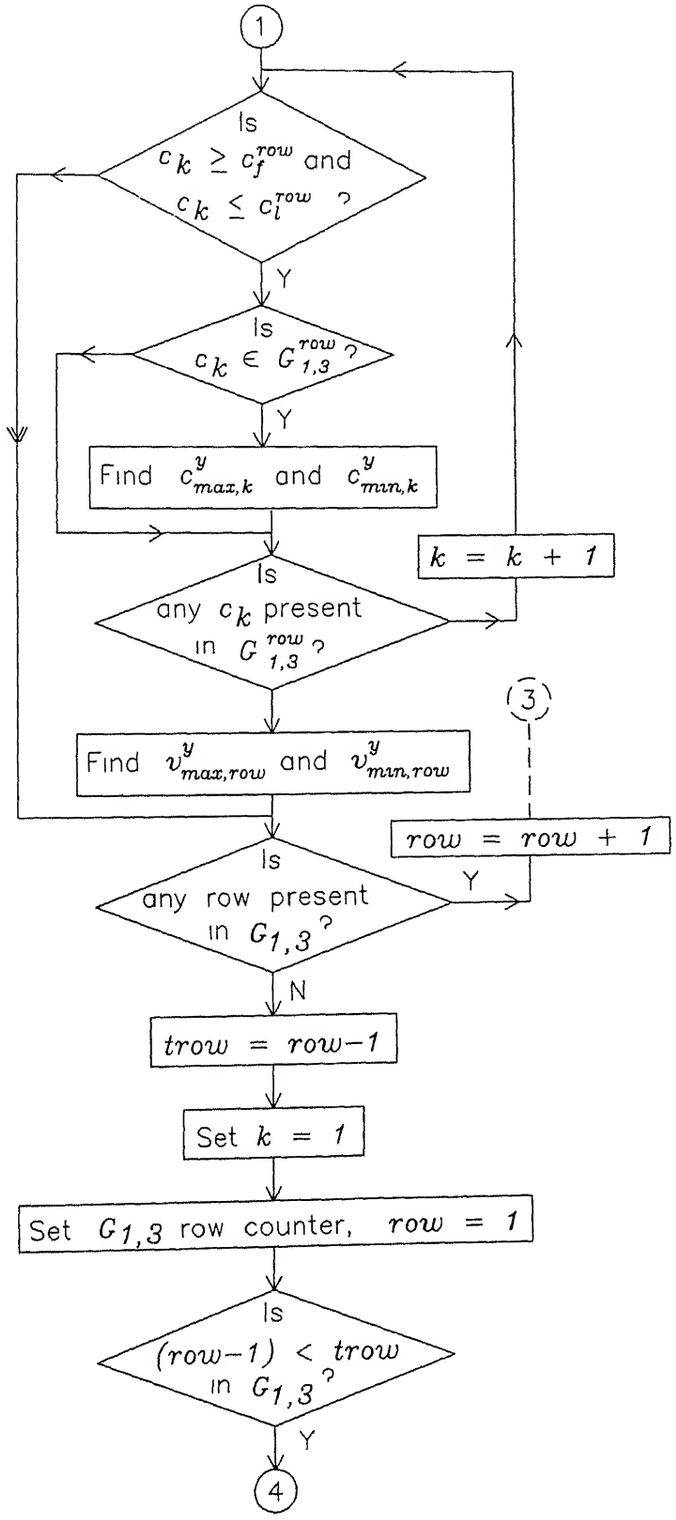


Figure 5.15: Flowchart for recognizing slitting features (continued).

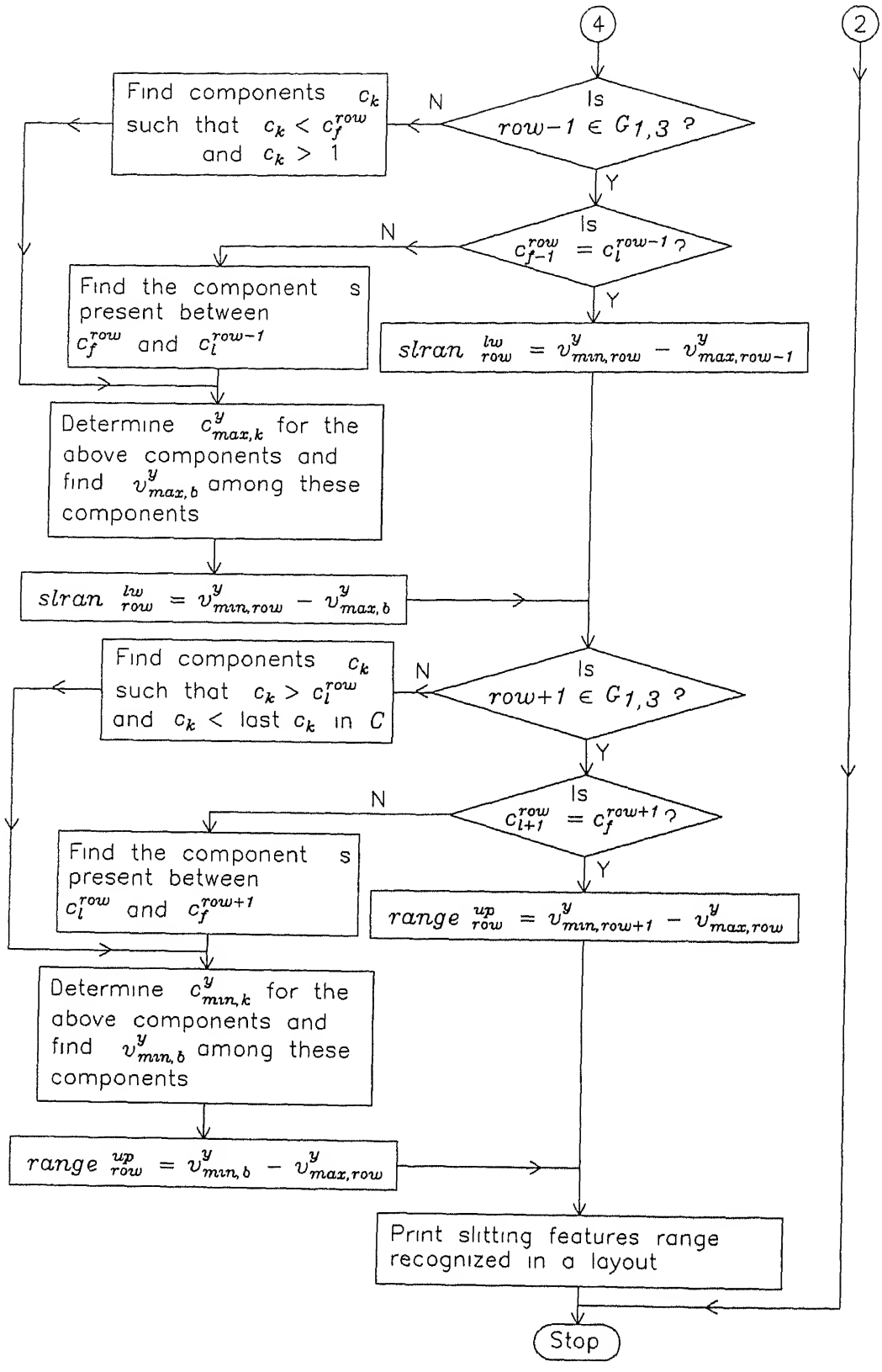


Figure 5.15· Flowchart for recognizing slitting features (continued)

slitting features (slf) :
units : mm

Sl. No.		x	y
1	sf of 1st row .		
	direction :	X axis	
	slr_{up} :	10 000	
	position :	30.000	245.000
		30.000	255 500

Table 5 13: Slitting features present in the component layout given in Fig 5.4.

for layout directions in '1' and '2' respectively From these subgraphs the component set $C_{1,3,5,7}$ and $C_{2,4,6,8}$ are determined

The shearing feature may be present in horizontal direction such that the raw material is of coil or strip type and is in the direction of Y axis, if raw material is of sheet type. The presence of shearing feature is determined as follows :

$$\text{If } C_{1,3,5,7} \text{ is } \begin{cases} \emptyset & \text{shearing feature is } \textit{absent} \\ \text{otherwise} & \text{shearing feature } \textit{may be present} \end{cases}$$

Similarly, whether the shearing feature is present or not in vertical direction is determined by examining the set $C_{2,4,6,8}$ which is given by

$$\text{If } C_{2,4,6,8} \text{ is } \begin{cases} \emptyset & \text{shearing feature is } \textit{absent} \\ \text{otherwise} & \text{shearing feature } \textit{may be present} \end{cases}$$

If the presence of shearing feature in horizontal direction is determined, then it is confirmed by the eqns. (5.18) and (5.19). Similarly shearing features in the vertical direction can be confirmed. After the confirmation of shearing features in the direction of X axis, their ranges (shr_{lw} and shr_{up}) are determined using eqns. (5.20) to (5.23). In the same way after the confirmation of shearing feature in the vertical direction, the ranges (shr_{rg} and shr_{lf}) are determined by the following procedure

For each column present in $G_{(2,4,6,8),col}$ the first component ($c_{f,col}$) and last component ($c_{l,col}$) are determined, and are given by :

$$c_{f,col} = \min_{k \in K} c_k \qquad \forall \quad c_k \in G_{(2,4,6,8),col} \tag{5.24(a)}$$

$$c_{l,col} = \max_{k \in K} c_k \qquad \forall \quad c_k \in G_{(2,4,6,8),col}. \tag{5.24(b)}$$

In the above equation, $G_{(2,4,6,8),col}$ is the col^{th} column of subgraph belonging to the $G_{2,4,6,8}$.

From these components present between the first and last component in a column, minimum and maximum x coordinate values for these components (i e., $(v^x_{min,k})$ and $(v^x_{max,k})$ respectively) are determined from the pseudo vertex set V'_j.

$$v^x_{min,k} = \min_{i \in I_j} v^x_{i,j} \quad | \ f_j \in B, \text{ and } c_k \in G_{(2,4,6,8),\,col} \tag{5 25(a)}$$

$$v^x_{max,k} = \max_{i \in I_j} v^x_j \quad | \ f_j \in B, \text{ and } c_k \in G_{(2,4,6,8),\,col}. \tag{5 25(b)}$$

Subsequently, minimum and maximum x coordinate values for each row (i.e., $v^x_{min,col}$ and $v^x_{max,col}$ respectively) present in the graph $G_{2,4,6,8}$ are determined.

$$v^x_{min,col} = \min_{c_k \in G_{(2,4,6,8),col}} v^x_{min,k} \quad \forall \ col \in G_{2,4,6,8} \tag{5.26(a)}$$

$$v^x_{max,col} = \max_{c_k \in G_{(2,4,6,8),col}} v^x_{max,k} \quad \forall \ col \in G_{2,4,6,8}. \tag{5 26(b)}$$

Shearing feature is present between two columns. The shearing range at the left side of the column (shr_{lf}) is given by the difference between the minimum x coordinate value of the column undr consideration $(v^x_{min,col})$, and maximum x coordinate value of the previous column $(v^x_{max,col-1})$ Similarly, shearing range at the right side of the column (shr_{rg}) is given by the difference between the minimum x coordinate value of the next column $(v^y_{min,col+1})$, and maximum x coordinate value of the column under consideration $(v^y_{min,col})$, i.e.,

$$shr_{lf} = v^x_{min,col} - v^x_{max,col-1} \tag{5.27(a)}$$

$$shr_{rg} = v^x_{min,col+1} - v^x_{max,col}. \tag{5.27(b)}$$

Thus, the shearing range is determined between columns. It should be noted that, if there are more than one column in a layout, then the shr_{rg} of the column under consideration and the next column's shr_{lf} will be same. Similarly, for component layout in direction '2', the presence of shearing features and their range are determined.

Implementation

In implementation, the column numbers are determined as per the identification process, which will be in lexicographical order. The output for Fig 5.7 is given in Table 5.14.

shearing features (shf)
units mm

Sl No.		x	y
1	shf of 1st row .		
	direction :	Y axis	
	shr_{rg} :	10.000	
	position :	80.500	174.000
		90 500	286.000
2	shf of 2nd row ·		
	direction :	Y axis	
	shr_{rg} :	10.000	
	position ·	120.500	174.000
		130.500	286.000

Table 5.14: Shearing features present in the component layout given in Fig. 5.7

Chapter 6

Forming Feature Recognition System

6.1 Introduction

Forming processes in sheet metal components are widely used in various types of industries like aerospace, automobiles, electronics, machine tools, etc , for manufacturing different types of components. In the forming processes, the work material is stressed beyond its elastic limit, but below the ultimate tensile strength Also, the 2–D work material is changed to 3–D components. Various forming operations used in different types of industries are given in Fig 2.2.

Nnaji et al., (1991), and Kang et al., (1993) have identified forming features from CAD database for components created by 3–D B_rep solid modeling. Forming features are identified based on the face oriented feature classification system, and the operations identified are bending, curling, shallow deep drawing, hemming, and louvering Hoffmann et al., (1993) have used feature based solid modeling for identifying bending features for press brake machines to sequence the bending operations on it

In the present research, forming features are recognized for components created in 3–D wireframe model representing the mean plane of the component. The standard DXF output file of AutoCAD Release–10 is used as input to present system. The raw material thickness and yield strength of the material are also given as input to the system. Forming features are identified from the component graph using set theoretic and graph theoretic concepts. The features are recognized based on the number of planes in which an operation is performed simultaneously, and location of the planes in the component. A set of rules have been developed by relating the features geometrically and topologically. The classification system proposed for recognizing forming features is given in Fig. 2.8.

In the forming feature recognition system of the present work, initially the component graph of a 3–D component created by wireframe model (which represents the mean plane of the component) is processed to identify parent and child subgraphs. Different loops present

in a subgraph are determined by vector normal method and vertex fusion methodology. Planes of a component are equivalent to the independent plane equations of loops present in the parent subgraph Also, the planes of a component in which child features along with child subgraph loops present are identified. Curved planes are identified, and adjacency plane relationship matrix is determined Cross-bend features are extracted from the component graph by removing common edges present between planes Subsequently, using the strategy of pattern recognition process, the extracted forming features are recognized by matching them with the features' graphs developed.

Flat pattern is developed for a 3–D component by determining bend allowance for curved planes by taking thinning effect into consideration Later, shearing features are identified from the flat pattern developed

6.2 Identification of parent and child subgraphs

Generation of a component graph from the CAD database is explained in Chapter 3. After the nature of component (i e., the 3–D component) from the graph is identified (explained in Section 3 7), then the component graph is processed to identify parent and child subgraphs Identification of parent and child subgraphs in turn helps in recognizing planes of a component and forming features

A component graph of a 3–D component consists of only one parent subgraph and child subgraphs may be absent or one or more than one child subgraph may be present in it. A set of principles are developed to relate the subgraphs of the component graph geometrically and topologically to determine the parent and child subgraphs. The mathematical model and the notations used are given below :

$a_{i,i'}$	boolean variable (0 or 1) of a vertex adjacency matrix of i^{th} row and i'^{th} column.
e_m	m^{th} edge of an edge set E, which is duplex, having start and end vertices as its elements, i.e.,$e_m = \{v_{s,m}, v_{e,m} \mid v_{s,m} \prec v_{e,m}\}$.
E	edge set of a component graph G, whose elements are lexicographically ordered, $E = \{e_m \mid m \in M\}$
E_j	edge set of j^{th} subgraph.
g_j	j^{th} element (subgraph) of a component graph G.
G	a 3–D component graph.
G_c	child subgraphs present in a component graph G.
G_p	parent subgraph present in a component graph G.
i	subscript of a vertex belonging to a vertex set V.
I	set of subscripts of a vertex set V.

I_j	set of subscripts of a vertex set V_j.
j	subscript of a subgraph belonging to a component graph G
J	set of subscripts of a component graph G
m	subscript of an edge belonging to an edge set E
M	set of subscripts of an edge set E
M_j	set of subscripts of an edge set E_j
len_j	projected length of j^{th} subgraph
v_i	i^{th} vertex of a vertex set V, which is a triplex having x, y, and z coordinates as its elements, i e , $v_i = \{x, y, z\}$
V	vertex set of a component graph G, whose elements are lexicographically ordered.
V_j	vertex set of j^{th} subgraph
x, y, z	represent the coordinate values of a vertex v.
X, Y, Z	represent X, Y, and Z axes respectively

The graph of a 3–D component G consists of subgraphs (g_j) and is given by

$$G = \{g_j \mid j \in J\} \tag{6 1}$$

where j is the subscript of graphs and J set of subscripts of subgraphs belonging to G.

The graph is made of parent subgraph (G_p) and child subgraphs (G_c), and hence the component graph (G) of a component is also given by

$$G = G_p \cup G_c \tag{6.2(a)}$$

where G_p and G_c are partitions of G, and satisfy the following condition :

$$\text{either} \quad G_p = G_c \quad \text{or} \quad G_p \cap G_c = \emptyset. \tag{6.2(b)}$$

The subgraphs of a component graph G are determined as given in Section 3.6 by the vertex fusion methodology along with the vertices and edges present in the subgraphs g_j and is given by

$$g_j = \{V_j, E_j\} \tag{6.3}$$

where V_j and E_j are vertex set and edge set of j^{th} subgraph The subscripts of vertices present in V_j are given by I_j and subscripts of edges present in E_j by M_j. Thus,

$$I_j = \{i \mid v_i \in V_j\} \tag{6.4}$$

and

$$M_j = \{m \mid e_m \in E_j\} \tag{6.5}$$

From the set G, firstly parent and child subgraphs are determined geometrically by finding maximum projected length (len_{max}) of subgraphs, and then relating them topologically

The maximum projected lengths of a subgraph along X, Y and Z directions are given by len^x_{max}, len^y_{max} and len^z_{max} respectively, and are determined as given below :

$$len^x_{max} = \max_{j \in J} len^x_j \tag{6 6(a)}$$

$$len^y_{max} = \max_{j \in J} len^y_j \tag{6 6(b)}$$

$$len^z_{max} = \max_{j \in J} len^z_j \tag{6.6(c)}$$

where len^x_j, len^y_j, and len^z_j represent the projected lengths of j^{th} subgraph in X, Y and Z directions respectively These are given by

$$len^x_j = v^x_{max,j} - v^x_{min,j} \tag{6.7(a)}$$

$$len^y_j = v^y_{max,j} - v^y_{min,j} \tag{6.7(b)}$$

$$len^z_j = v^z_{max,j} - v^z_{min,j} \tag{6.7(c)}$$

$$\forall \quad j \in J.$$

In eqn. (6.7(a)), $v^x_{max,j}$ and $v^x_{min,j}$ represent maximum and minimum x coordinate values of j^{th} subgraph respectively, and are determined as given below .

$$v^x_{max,j} = \max_{i \in I_j} v^x_{ij} \tag{6.8(a)}$$

$$v^x_{min,j} = \min_{i \in I_j} v^x_{ij} \tag{6.8(b)}$$

$$\forall \quad j \in J,$$

where v^x_{ij} represents the x coordinate value of the i^{th} vertex of j^{th} subgraph, and I_j represents the set of subscripts of vertices present in vertex set V_j. The elements (vertices) of V_j are present in a subgraph g_j of the component graph G.

Similarly, maximum and minimum y and z coordinate values of vertices of j_{th} subgraph ($v^y_{max,j}$ and $v^y_{min,j}$; and $v^z_{max,j}$ and $v^z_{min,j}$) are determined by the following equations :

$$v^y_{j,max} = \max_{i \in I_j} v^y_{ij} \tag{6.9(a)}$$

$$v^y_{j,min} = \min_{i \in I_j} v^y_{ij} \tag{6.9(b)}$$

$$\forall \quad j \in J$$

and

$$v^z_{max,j} = \max_{i \in I_j} v^z_{ij} \quad (6\ 10(a))$$

$$v^z_{min,j} = \min_{i \in I_j} v^z_{ij} \quad (6\ 10(b))$$

$$\forall \quad j \in J$$

Subgraphs having maximum projected lengths in X, Y, and Z directions (G_X, G_Y, and G_Z) are given by the following equations

$$G_X = \{g_j \mid len^x_j = len^x_{max}\} \quad (6\ 11(a))$$

$$G_Y = \{g_j \mid len^y_j = len^y_{max}\} \quad (6.11(b))$$

$$G_Z = \{g_j \mid len^z_j = len^z_{max}\} \quad (6.11(c))$$

$$\forall \quad j \in J.$$

6.2.1 Identification of parent subgraph

A parent subgraph is defined as a component of the graph that lies outside all other subgraphs. Therefore, topologically the parent subgraph is the outer most subgraph. Hence, from eqn. (6.11), if the set G_{XYZ} (eqn (6.12)) is not an empty or null set, then the element of the set (G_{XYZ}) represents the parent subgraph of a component graph, and is given by

$$G_{XYZ} = G_X \cap G_Y \cap G_Z \quad (6.12)$$

or

$$G_p = \{g_j \mid g_j \in G_{XYZ}\} \quad (6\ 13)$$

It should be noted that the set G_{XYZ} in eqn (6.12) is either a single element set or a null set. If G_{XYZ} is a null set, then the subgraphs having maximum projected length in at least two directions (i.e., X and Y, or Y and Z, or Z and X) are determined. Let G_{XY}, G_{YZ}, and G_{ZX} be set of subgraphs having maximum projected lengths in two directions viz., in X and Y, Y and Z, and Z and X respectively, and are given by

$$G_{XY} = G_X \cap G_Y \quad (6\ 14(a))$$

$$G_{YZ} = G_Y \cap G_Z \quad (6\ 14(b))$$

$$G_{ZX} = G_Z \cap G_X \quad (6\ 14(c))$$

The parent subgraph G_p is identified from the eqn. (6.14), where only one of the set will be a single element set and others will be null set, i.e.,

$$G_p = \{g_j \mid g_j \in G_{XY}, \text{ or } g_j \in G_{YZ}, \text{ or } g_j \in G_{XY}\}. \quad (6.15)$$

6.2.2 Identification of child subgraphs

A child subgraph is defined as a component of the graph and it lies inside the parent subgraph Therefore, topologically a child subgraph always lies inside another subgraph. Since, parent and child subgraphs are partition sets of G (in accordance to eqn. (6.2(a))), the component graph set G can be written as

$$G = G_p + G_c \tag{6 16}$$

Hence, child subgraphs are determined by the eqn (6 17)

$$G_c = G - G_p. \tag{6.17}$$

Child subgraph set (G_c) can also be stated as

$$G_c = \{g_j \mid g_j \in G, g_j \notin G_p, j \in J\} \tag{6.18}$$

Implementation

Flowchart for identifying parent and child subgraphs is shown in Fig. 6.1 Input to the system are vertices and edges present in each subgraph. An input to the system for the component shown in Fig. 3.2 is given in Fig 3.12. The identified parent and child subgraphs are given in Table 6.1. Also, the subgraphs having maximum projected lengths in X, Y, and Z directions (G_X, G_Y, and G_Z), and in all the three directions simultaneously (G_{XYZ}) are given in Fig (6.2) The output (Table 6.1) indicates that the component has the maximum projected length in all three directions Another component having maximum projected length in X and Y directions is shown in Fig. 6.2(a), and its graph in Fig. 6.2(b). The subgraph along with its vertices and edges obtained from the connectedness and subgraph identification module, are given in Table 6.2(a). With this as input to the system, its output is given in Table 6.2(b).

6.3 Identification of loops in a component graph

A 3–D component and its graph are shown in Figs. 6.3 (a) and 6.3(b) respectively. Various subgraphs along with their vertices and edges (Table 6.3) are determined according to the procedure given in Section 3.6. Recognition process for parent (g_1) and child subgraphs (g_2 and g_3 in Table 6.3) of a component graph, has been explained in the previous section. Parent and child subgraphs are made up of loops. A loop in the parent subgraph represents a plane in a component. If there are more than one loop in a child subgraph, then at least one of the loops will be located inside a loop of the parent subgraph, and other loops represent different planes in a component. On the other hand, if there is only one loop in a child subgraph, then it lies inside a loop of the parent subgraph.

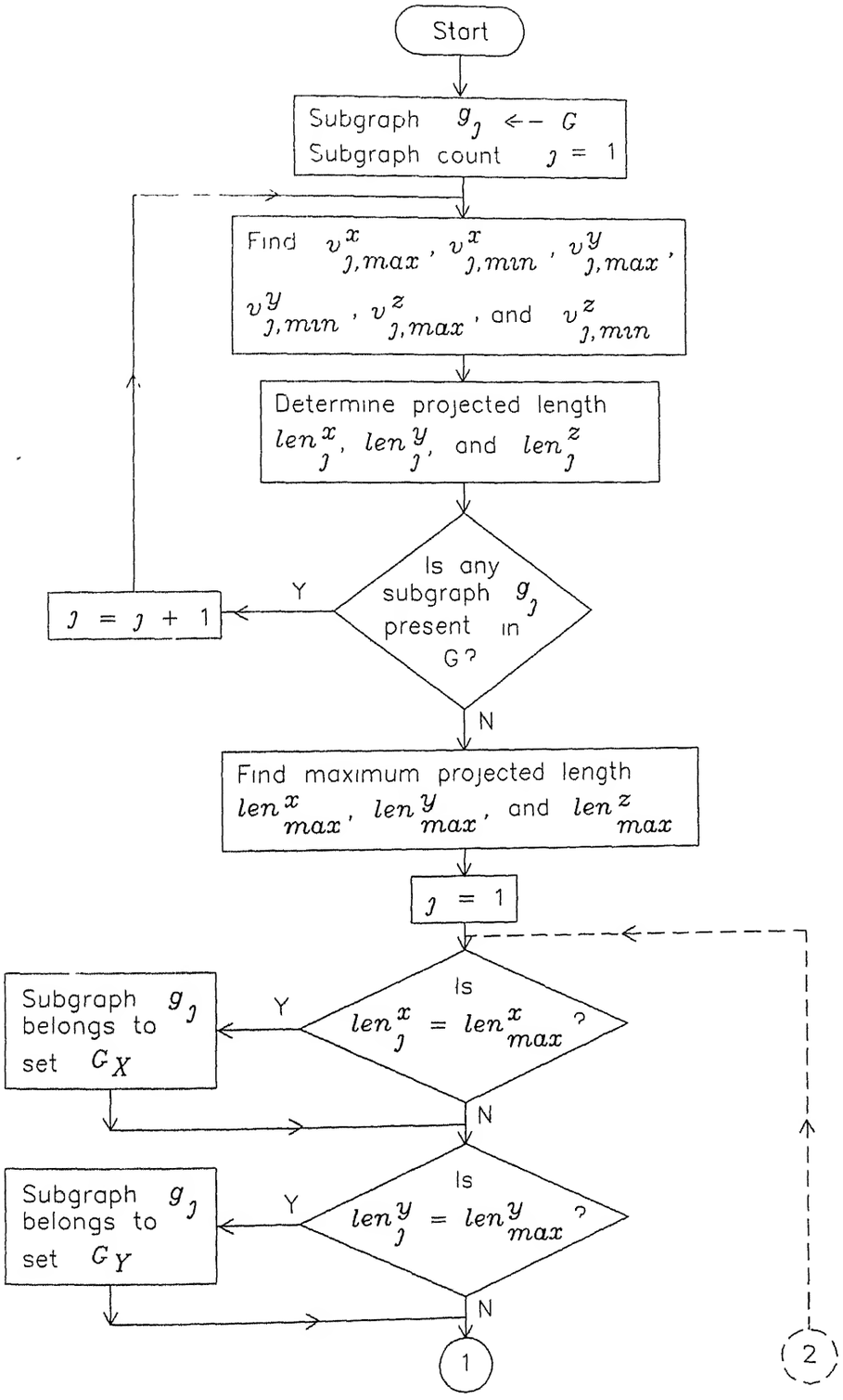


Figure 6.1: Flowchart for identifying parent and child subgraphs

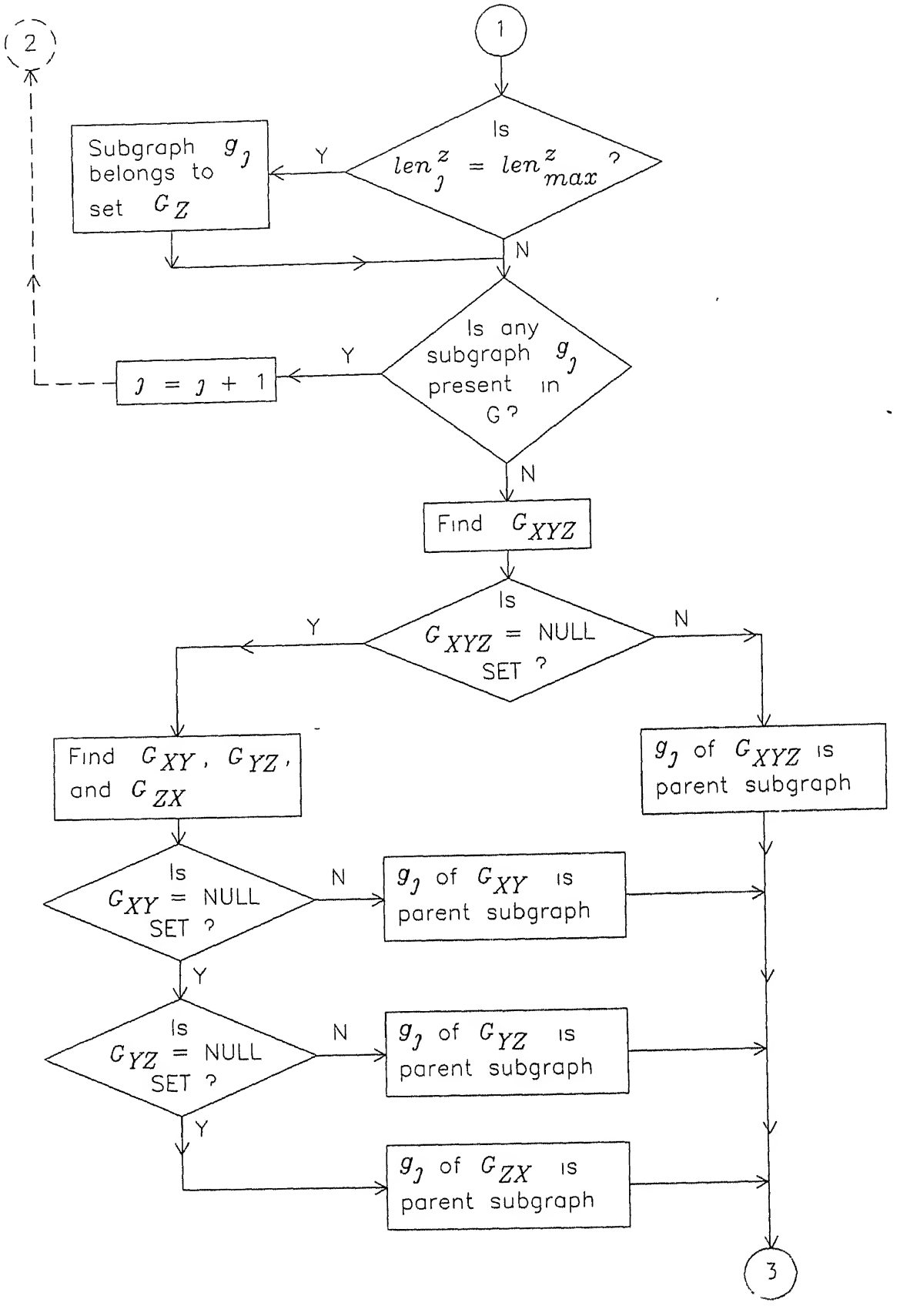


Figure 6.1: Flowchart for identifying parent and child subgraphs (continued).

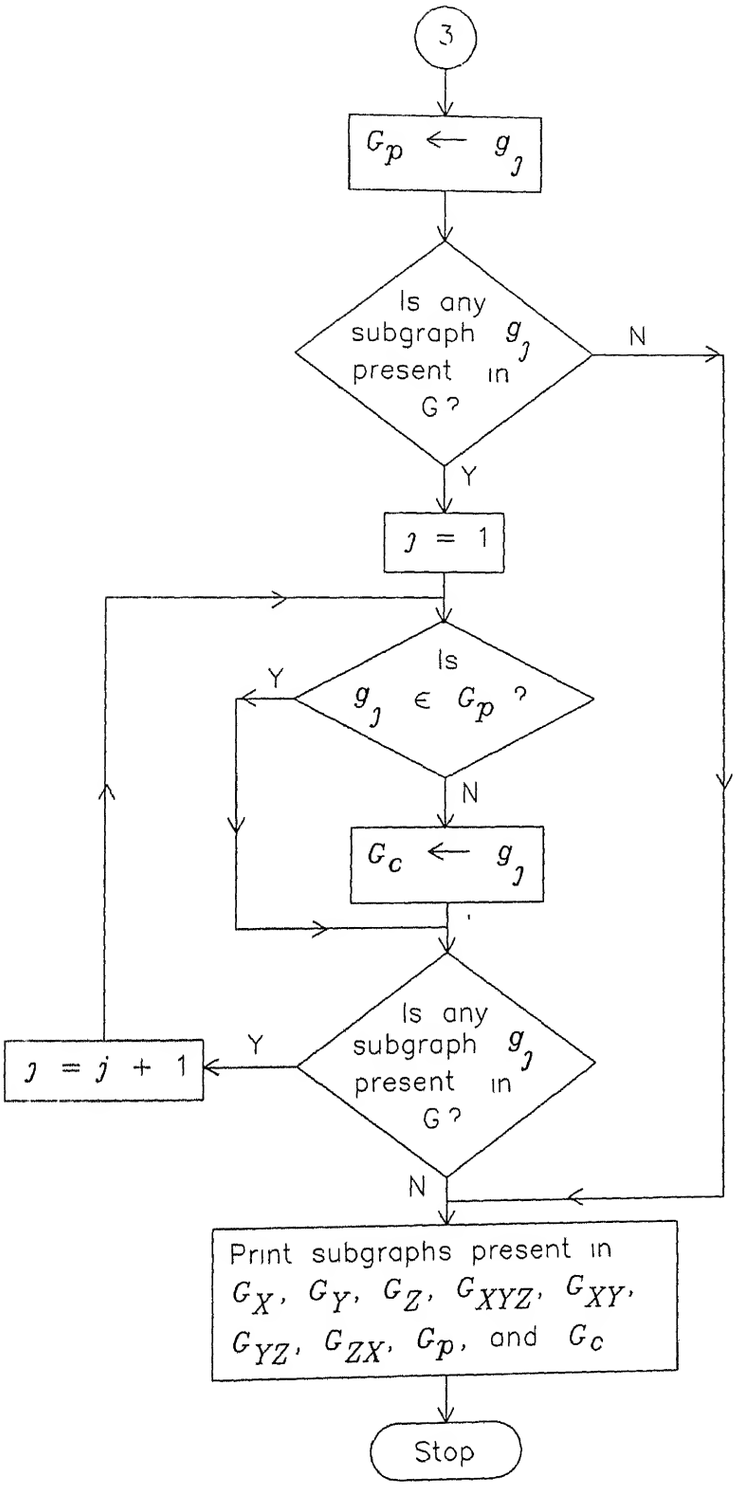


Figure 6.1· Flowchart for identifying parent and child subgraphs (continued).

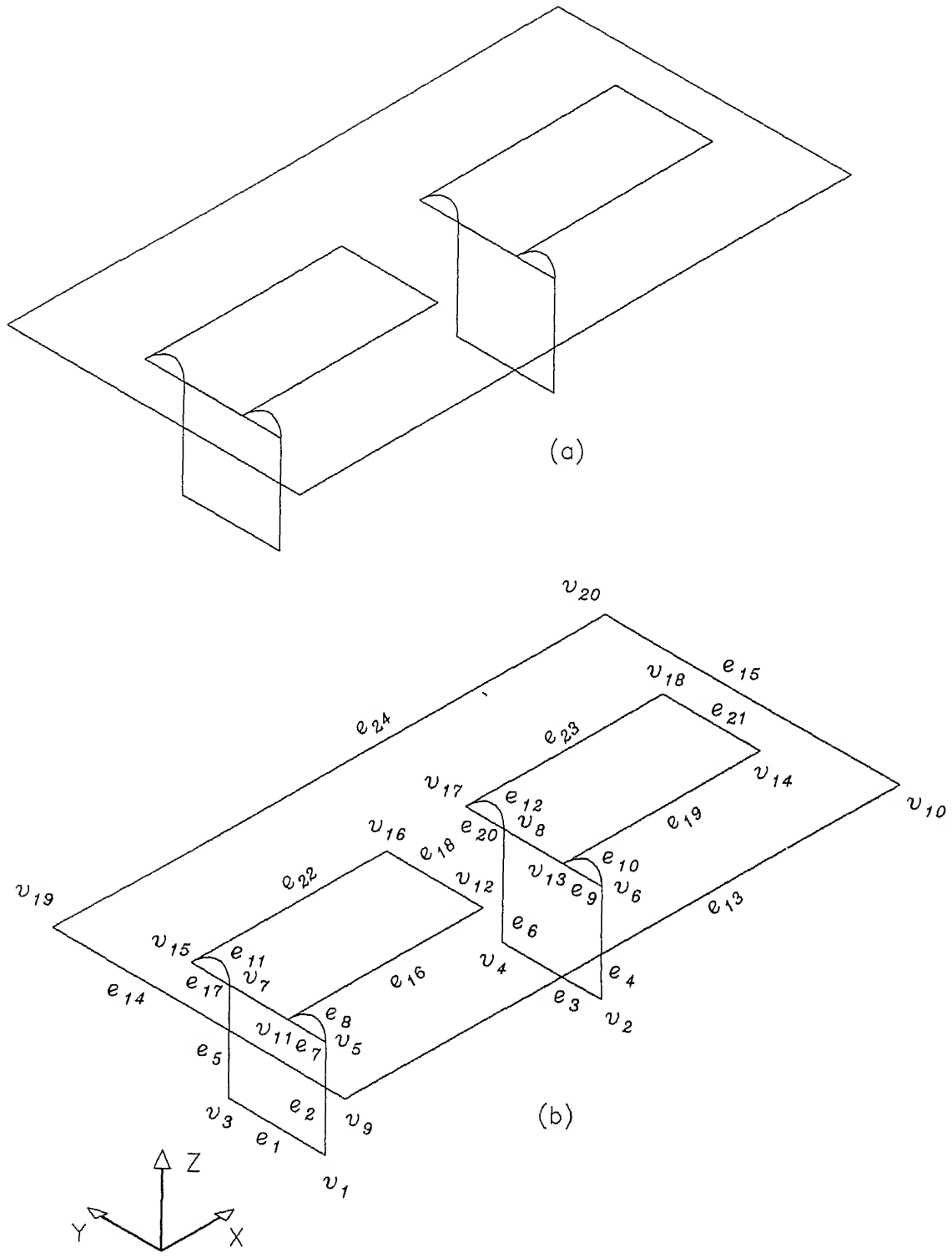


Figure 6.2: A 3–D component having maximum projected length in two directions : (a) Component, and (b) Component graph with lexicographically ordered vertices and edges.

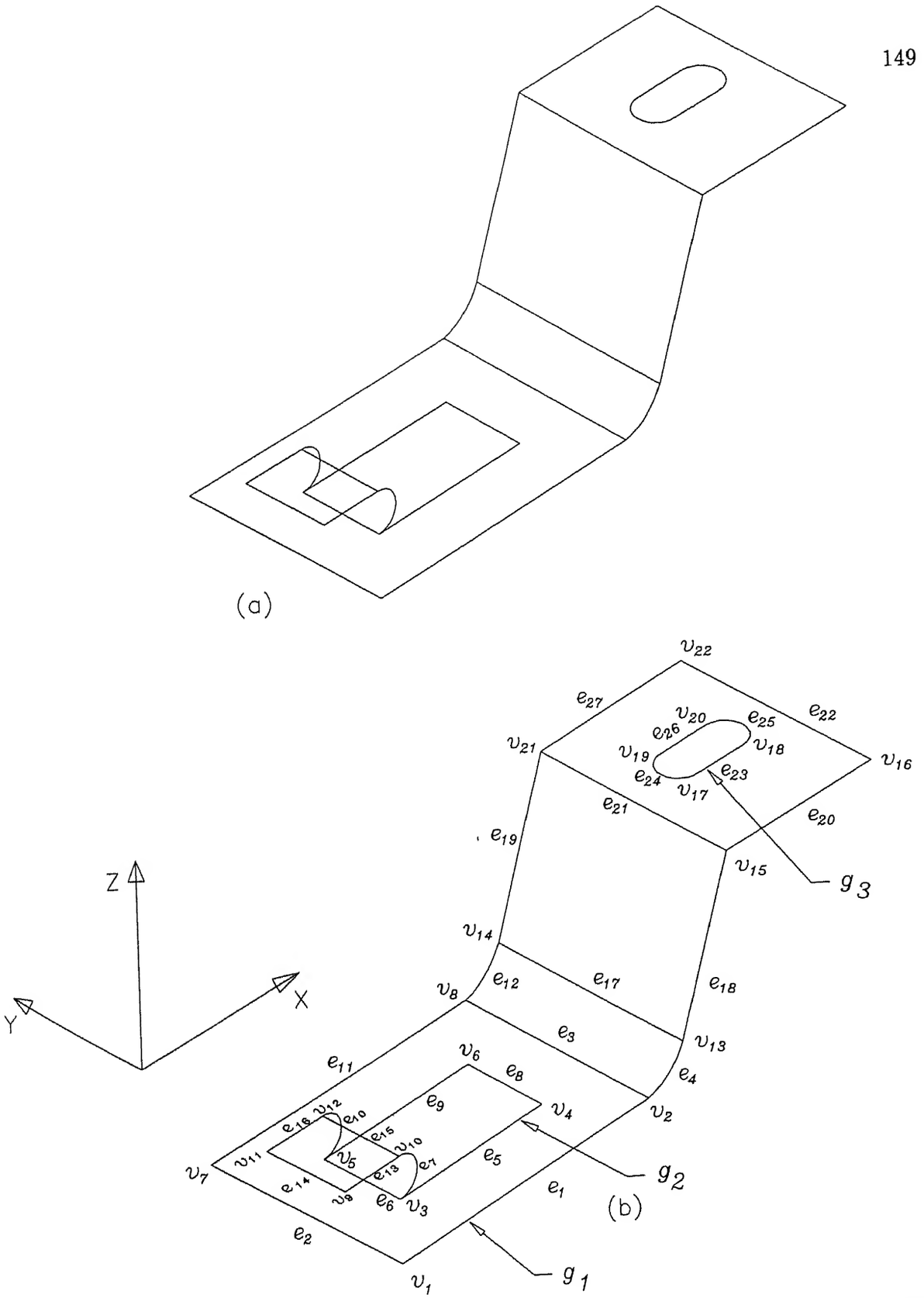


Figure 6.3: A 3–D component having 3–D and 2–D child subgraph features (a) Component, and (b) Component graph with lexicographically ordered vertices and edges.

Subgraphs	G_X	g_1	
Subgraphs	G_Y	g_1	
Subgraphs	G_Z	g_1	
Subgraphs	G_{XYZ}	g_1	
Subgraphs	G_p	g_1	
Subgraphs	G_c	g_2	g_3

Table 6.1· Parent and child subgraphs of a component graph having the maximum projected length in all three directions

6.3.1 Determination of number of independent loops in a component graph

The number of loops present in a component graph is determined in accordance with Euler's formula for wireframe model, and is given below :
Let

L_G loops present in the component graph.
L_j loops present in the j^{th} subgraph
S models or subgraphs present in the component graph.

According to Euler's formula for wireframe model (eqn.(3.1)), the number of loops (L_G) in a graph is given by

$$L_G = 2S + E - V. \tag{6.19}$$

In Fig. 6 3(b), number of loops present in the component (L_G) is $(2 \times 3 + 27 - 22) = 11$, and is calculated by Maze algorithm (Wilson, 1985). However, the number of independent loops (L) which represent planes, or forming features, or both in a component, is given by

$$L = L_G - S. \tag{6 20}$$

In Fig. 6.3.(b), L is $11 - 3 = 8$.

Identification of independent loops in parent and child subgraphs of a component graph and the loop of the parent subgraph in which the loop of child subgraph is located are discussed in the following sections.

Subgraph	g_1	v_1 v_3 v_5 v_7 v_{11} v_{12} v_{15} v_{16}
Subgraph	g_2	v_2 v_4 v_6 v_8 v_{13} v_{14} v_{17} v_{18}
Subgraph	g_3	v_9 v_{10} v_{19} v_{20}

Subgraph	g_1	e_1 e_2 e_5 e_7 e_8 e_{11} e_{16} e_{17} e_{18} e_{22}
Subgraph	g_2	e_3 e_4 e_6 e_9 e_{10} e_{12} e_{19} e_{20} e_{21} e_{23}
Subgraph	g_3	e_{13} e_{14} e_{15} e_{24}

(a)

Subgraphs	G_X	g_3
Subgraphs	G_Y	g_3
Subgraphs	G_Z	g_1 g_2
Subgraphs	G_{XYZ}	NULL
Subgraphs	G_{XY}	g_3
Subgraphs	G_{YZ}	NULL
Subgraphs	G_{ZX}	NULL
Subgraphs	G_p	g_3
Subgraphs	G_c	g_1 g_2

(b)

Table 6.2: For Fig (6 2) : (a) Vertices and edges of subgraphs present in the component graph, and (b) Parent and child subgraphs of a component graph having maximum projected length in X and Y directions

Subgraph	g_1	v_1 v_2 v_7 v_8 v_{13} v_{14} v_{15} v_{16} v_{21} v_{22}
Subgraph	g_2	v_3 v_4 v_5 v_6 v_9 v_{10} v_{11} v_{12}
Subgraph	g_3	v_{17} v_{18} v_{19} v_{20}
Subgraph	g_1	e_1 e_2 e_3 e_4 e_{11} e_{12} e_{17} e_{18} e_{19} e_{20} e_{21} e_{22} e_{27}
Subgraph	g_2	e_5 e_6 e_7 e_8 e_9 e_{10} e_{13} e_{14} e_{15} e_{16}
Subgraph	g_3	e_{23} e_{24} e_{25} e_{26}
Subgraphs	G_p	g_1
Subgraphs	G_c	g_2 g_3

Table 6.3: For the 3-D component shown in Fig. 6 3, vertices and edges of subgraphs present in the various subgraphs, and parent and child subgraphs, present in the component are given.

6.3.2 Calculation and identification of loops in a parent subgraph

Determination of number of loops and number of independent loops in a parent subgraph is similar to that of the identification of number of loops and independent loops in a subgraph

Calculating the number of loops and independent loops present in a parent subgraph

The number of loops (L_{Gp}), and the number of independent loops (L_p) present in the parent subgraph of a component graph are determined with the help of eqns. (6 21) and (6 22) after finding the number of vertices and edges present in the subgraph.

$$L_{Gp} = 2S + E_j - V_j, \tag{6 21}$$

where E_j and V_j are edge set and vertex set of j^{th} subgraph belonging to the parent subgraph set G_p.

$$L_p = L_{Gp} - S \tag{6 22}$$

Fig 6.3(a) shows a 3–D component and Fig 6 3(b) shows its graph Vertices and edges present in the parent subgraph (g_1), determined by vertex fusion methodology are given in Table 6.3. There are 10 vertices, and 13 edges in the subgraph g_1. Therefore, the number of loops (L_{Gp}) present in the subgraph g_1 is determined as 5.

$L_{Gp} = 2 \times 1 + 13 - 10 = 5.$

Number of independent loops (L_p) present in the parent subgraph (g_1) is evaluated by eqn (6.22), which is given by

$L_p = 5 - 1 = 4.$

Identification of loops in the parent subgraph

Loops present in a parent subgraph are identified from edges and vertices present in it by vector normal method and vertex fusion methodology. By vector normal method (discussed in Section 3.7), a plane equation is determined, and various edges of the parent subgraph present in the plane under consideration are identified. Edges present in the plane are further processed to identify different loops present in the plane by the principle of vertex fusion methodology (explained in Section 3 6).

Methodology

Initially edge adjacency (EA) matrix is determined for a parent subgraph. EA matrix of the parent subgraph g_1 (Fig. 6.4(a)) is shown in Fig. 6 4(b) Edges in the EA matrix are placed in the same order as that in the edge set E_1. If the edges are adjacent to each

other, then the value assigned to the cell is '1', else '0' ('0' is represented by an empty cell in EA matrix), i e ,

$$a_{m,m'} = \begin{cases} 1 & \text{if edges } e_m \text{ and } e_{m'} \text{ are adjacent to each other} \\ 0 & \text{otherwise } (j \neq j') \end{cases}$$

$$e_m \in E_j, e_{m'} \in E_j,$$

$$e_m \neq e_{m'}, \text{ and } E_j \in g_j$$

where $a_{m,m'}$ indicates the value of m^{th} row and m'^{th} column of the EA matrix. Edges e_m and $e_{m'}$ are elements of lexicographically ordered set E_j, which is a subset of lexicographically ordered edge set E. Hence, E_j is also ordered lexicographically An edge e_m *is said to be adjacent to another edge* $e_{m'}$, when one of the vertices is common to both edges.

Edges present in the parent subgraph g_1 are given in Table 6.3. They are elements of edge set (E_1) and are given by

$E_1 = \{e_1, e_2, e_3, e_4, e_{11}, e_{12}, e_{17}, e_{18}, e_{19}, e_{20}, e_{21}, e_{22}, e_{27}\}$.

The degree of an edge $d(e_m)$ for edges present in the EA matrix is determined as given in eqn. (6.23) and is defined as the number of edges adjacent to it.

$$d(e_m) = \sum_{\forall m} a_{m,m'} \qquad \forall \quad m'. \tag{6.23}$$

Various edges can be a part of a loop, if and only if the edge degree $(d(e_m))$ is greater than zero It should also be noted that, if $d(e_m)$ of a graph is zero, then the m^{th} edge represents a self loop obtained for circular entity. Also, edges can be common for utmost two loops in the graphs of a sheet metal component. The maximum number of loops that an edge can have in common (loop degree of an edge $(l(e_m))$ is determined as given below :

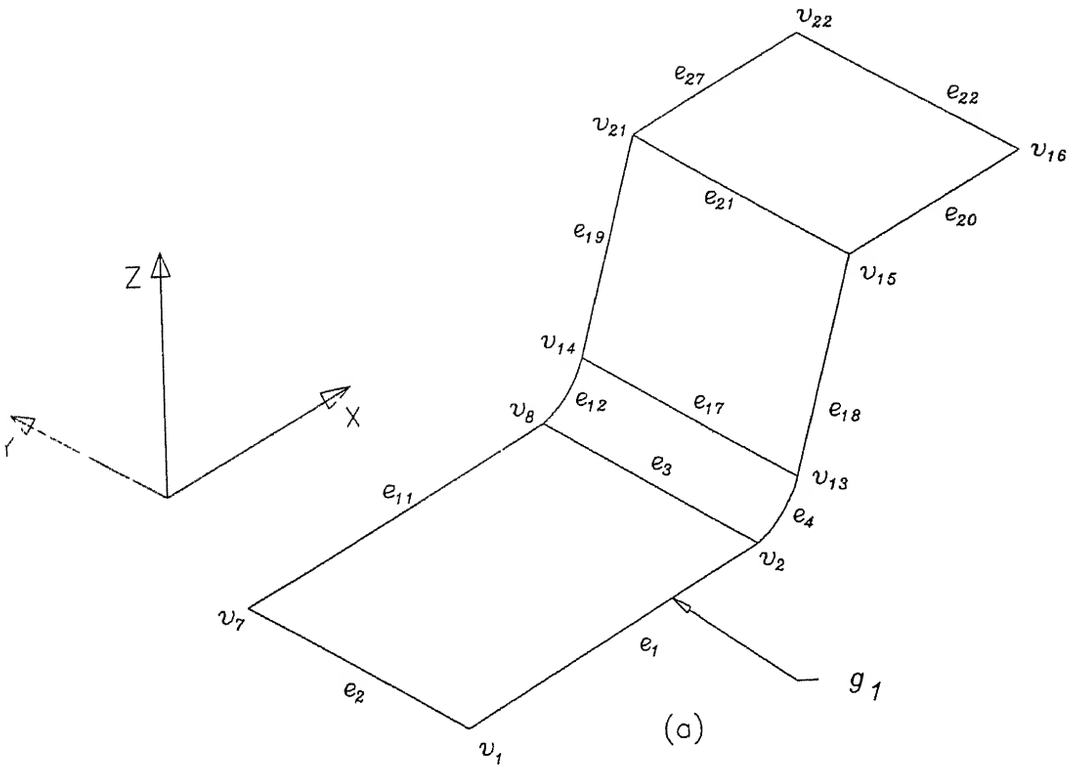
$$l(e_m) = min(d(v_{s,m}), d(v_{e,m})) - 1. \tag{6 24}$$

For the parent subgraph (g_1) under consideration, $d(e_m)$ and $l(e_m)$ are given in Fig. 6 4(b).

If $d(e_m)$ of at least one edge present in EA matrix is greater than one, then the independent loops in the subgraph are identified. The loops are identified by determining edge vectors (eqn. (3 14)) for a pair of edges in the edge set E_j, such that they are adjacent to each other, and they precede all other adjacent edges in E_j.

For EA matrix shown in Fig 6.4(b), since $d(e_m)$ of edges are greater than one, first two edges e_1 and e_2 are selected, as they are adjacent to each other, and they precede all other edges in the edge set E_1. The plane equation is determined by finding a vector normal $(\vec{NV})$ to the edge vectors as discussed in section 3.7 and is given by

$$NV^x_{m,m'}x + NV^y_{m,m'}y + NV^z_{m,m'}z = CV_{m,m'} \tag{6 25}$$



e_k \ $e_{k'}$	1	2	3	4	11	12	17	18	19	20	21	22	27	$d(e_k)$	$l(e_k)$
1	–	1	1	1										3	1
2	1	–			1									2	1
3	1		–	1	1	1								4	2
4	1		1	–			1	1						4	2
11		1	1		–	1								3	1
12			1		1	–	1		1					4	2
17				1		1	–	1	1					4	2
18				1			1	–		1	1			4	2
19						1	1		–		1		1	4	2
20								1		–	1	1		3	1
21								1	1	1	–		1	4	2
22										1		–	1	2	1
27									1		1	1	–	3	1

(b)

Figure 6.4: For the 3–D component given in Fig. 6 3(a) (a) Parent subgraph of the component, and (b) Edge adjacency (EA) matrix of parent subgraph along with degree of an edge and degree of a loop.

where the coefficients of x, y, and z are coefficients of unit vectors i, j, and k of the normal vector ($\vec{NV}$) respectively The constant $CV_{m,m'}$ in eqn (6.25) is determined by substituting the coordinate values of any one of the vertex belonging to edges e_m or $e_{m'}$, since they are located in the same plane.

Edges other than the edges e_m and $e_{m'}$ of the subgraph are processed to determine its presence in the plane under consideration (An edge is said to be contained in the plane, if and only if both start and end vertices are lying in the plane under consideration Start and end vertices are said to be in the plane, if both vertices satisfy the plane equation (eqn. (6 25)) The process of determining edges in a plane is continued till no more edges are left in the edge set E_j

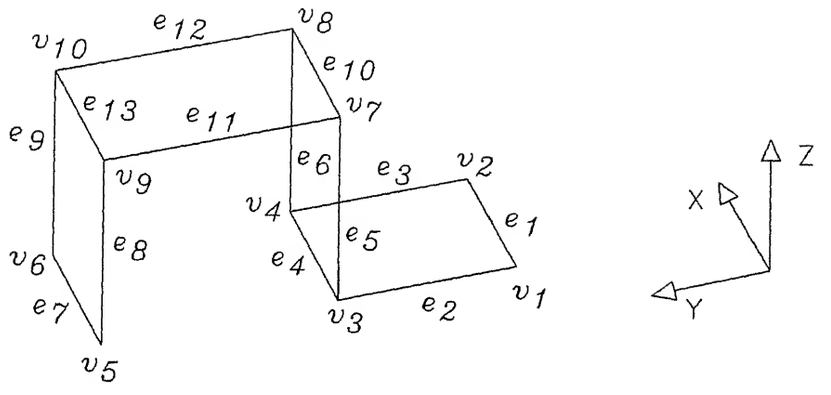
In the process of identifying edges of the plane under consideration, all edges that are identified may not be present in a single loop or they may form a path. Therefore, connectedness of edges in a loop is determined by vertex fusion methodology

In Fig. 6.5(a) edges e_1, e_2, e_3, e_4, and e_7 lie in the same plane, but only e_1, e_2, e_3, and e_4 form a loop. Therefore, to find the connectedness between edges in a plane, vertex fusion methodology is used. The output of vertex fusion methodology for the plane under consideration is given in Fig. 6 5(b)

After identifying a loop, the loop count ($lc(e_m)$) of edges forming a loop is incremented by one If the loop count of edges and their loop degree ($l(e_m)$) are equal, then rows and columns of those edges are discarded from the matrix. If all the rows and columns get discarded from the EA matrix, then it represents a single loop in the parent subgraph, and hence only one plane is present in it Otherwise two more edges forming a different loop or path are selected, such that they are adjacent to each other, and precede other adjacent pair of edges in the EA matrix. If the edges selected provide a path, but not the loop, then different edges are chosen retaining the previous selected edges in the EA matrix. The process is continued till the edges in the remaining EA matrix do not provide a loop. In the EA matrix, edges do not provide a loop, if $d(e_m)$ of at least one edge is not greater than one. This methodology is explained below for the parent subgraph g_1.

Implementation

The EA matrix for the parent subgraph (g_1) is given in Fig. 6.4(b). Initially two edges e_1 and e_2 are selected, and the plane equation is determined. Other edges which satisfy this plane equation are e_3 and e_{11}. By vertex fusion methodology, it is found out that all the four edges form a single loop. Therefore, the loop count of edges ($lc(e_m)$) is incremented by one. After the increment, the loop count and loop degree of e_1, e_2 and e_{11} are found to be equal, i.e., one. Therefore, in Fig. 6.6(a), the EA matrix along with the degree of edges



Edges in a plane p_1 e_1 e_2 e_3 e_4 e_7

(a)

Edges in a loop l_1 e_1 e_2 e_3 e_4

Edges in a path l_2 e_7

(b)

Figure 6.5: A 3–D component having edges in the same plane, but forming a loop and a path : (a) Graph showing edges e_1, e_2, e_3, e_4 and e_7 located in a plane, and (b) Connectedness of edges in the plane under consideration

e_m / $e_{m'}$	3	4	12	17	18	19	20	21	22	27	$d'(e_m)$	$l(e_m) - lc(e_m)$
3	–	1	1								2	1
4	1	–		1	1						3	2
12	1		–	1		1					3	2
17		1	1	–	1	1					4	2
18		1		1	–		1	1			4	2
19			1	1		–		1		1	4	2
20					1		–	1	1		3	1
21					1	1	1	–		1	4	2
22							1		–	1	2	1
27						1		1	1	–	3	1

(a)

e_m / $e_{m'}$	4	12	17	18	19	20	21	22	27	$d'(e_m)$	$l(e_m) - lc(e_m)$
4	–		1	1						2	1
12		–	1		1					2	1
17	1	1	–	1	1					4	1
18	1		1	–		1	1			4	2
19		1	1		–		1		1	4	2
20				1		–	1	1		3	1
21				1	1	1	–		1	4	2
22						1		–	1	2	1
27					1		1	1	–	3	1

(b)

e_m / $e_{m'}$	4	12	18	19	20	21	22	27	$d'(e_m)$	$l(e_m) - lc(e_m)$
4	–		1						1	1
12		–		1					1	1
18	1		–		1	1			3	1
19		1		–		1		1	3	1
20			1		–	1	1		3	1
21			1	1	1	–		1	4	1
22					1		–	1	2	1
27				1		1	1	–	3	1

(c)

e_m / $e_{m'}$	4	12	18	19	$d'(e_m)$	$l(e_m) - lc(e_m)$
4	–		1		1	1
12		–		1	1	1
18	1		–		1	1
19		1		–	1	1

(d)

Figure 6.6: Steps in determining independent loops present in the parent subgraph.

($d'(e_m)$) after discarding the rows and columns representing e_1, e_2, and e_{11} is given in Fig. 6.4(b).

Since, the EA matrix is not empty and $d'(e_m)$ of few edges is greater than one, the identification process of edges in a plane is continued

Now, the edges e_3 and e_4 are selected for determining edge vectors, and in turn the plane equation. The edges in EA matrix which also satisfy this plane equation are e_{12} and e_{17}. By vertex fusion methodology, it is found out that all these four edges form a loop As before, the $lc(e_m)$ of edges is incremented by one. Since loop count of e_3 and its loop edge are equal (i.e., $(l(e_m) - lc(e_m)) = 0$), the row and column representing it are deleted from the matrix. The remaining EA matrix along with $d'(e_m)$ and ($l(e_m)$ - $lc(e_m)$) are shown in Fig. 6.6(b).

In EA matrix, $d'(e_m)$ of few edges is greater than one, and hence loop identification process is continued. From the EA matrix of Fig. 6.6(b), a pair of edges e_4 and e_{17} is selected, which are adjacent to each other and the pair precedes all other adjacent pairs of

edges in the edge set E_j

After determining the plane equation and also checking for edges contained in this plane, it is found out that the edge e_{12} is located in the plane Edges e_4, e_{12} and e_{17} are connected (determined by vertex fusion methodology), but they form a path Therefore, next pair of edges (e_4 and e_{18}) adjacent to each other is selected as the pair precedes other adjacent pair of edges in the edge set E_1 Forming of plane equation and checking for edges lying in the plane are continued, and it is found out that there is no other edge which lies in this plane, and hence e_4 and e_{18} also forms a path During the subsequent selection of pair of edges to determine plane equations (e_{12} and e_{17} and e_{12} and e_{19}), loops are not determined, but still loop identification process is continued, since $d'(e_m)$ of all the edges is greater than '1' (Fig. 6.6(b)

When e_{17} and e_{18} are selected to find the plane equation, a loop is determined by having e_{17}, e_{18}, e_{19} and e_{21} as its edges Therefore, the loop count of these edges is incremented by one. The row and column representing e_{17} are deleted from EA matrix, and the remaining EA matrix is shown in Fig 6.6(c).

Edge degree $d'(e_m)$ of few edges is greater than '1' in EA matrix, and therefore the loop identification process is continued and another loop having edges e_{20}, e_{21}, e_{22} and e_{27} is determined. After identification of this loop, the EA matrix is shown in Fig. 6.6(d).

In this EA matrix of parent subgraph, the edge degree ($d'(e_m)$) of the all edges is one (which indicates that every edge has only one edge as its adjacent edge), and therefore they cannot form a loop. Hence loop finding process is stopped.

It should be noted that, the edges remaining in the EA matrix (Fig 6.6(d)) form a part of boundary edges of the component. Edges having loop degree of '2', and which are not present in the final EA matrix are common edges between the two loops Thus e_4, e_{12}, e_{18} and e_{19} form a part of boundary loop of the component graph and e_3, e_{17} and e_{21} are common edges present in two different loops of the parent subgraph (g_1). The edges which are identified in different loops are given in Table 6.4. It can be seen that the number of independent loops identified in the g_1 are 4 ($l_{g1,1}$, $l_{g1,2}$, $l_{g1,3}$, $l_{g1,4}$) which is in accordance with eqn (6.22).

During implementation discarding rows and columns is a cumbersome process, and therefore flags are used for recognizing whether to consider the particular row and column or not for further processing. The output of edges in each loop, and common edges present in the subgraph are given in Table 6.4.

Loop in a parent subgraph	$l_{g1,1}$	e_1 e_2 e_3 e_{11}
Loop in a parent subgraph	$l_{g1,2}$	e_3 e_4 e_{12} e_{17}
Loop in a parent subgraph	$l_{g1,3}$	e_{17} e_{18} e_{19} e_{21}
Loop in a parent subgraph	$l_{g1,4}$	e_{20} e_{21} e_{22} e_{27}
Common edges		e_3 e_{17} e_{21}

Table 6 4· Edges present in different loops of the parent subgraph, and common edges present in various loops

6.3.3 Identification of loops in child subgraphs

Number of loops (L_c) present in child subgraphs of a component graph is determined as below :

$$L_c = L - L_p. \tag{6 26}$$

In case of a component graph (Fig. 6.3(b)), the number of independent loops present in child subgraphs are 4 ($L_c = 8 - 4 = 4$). If there are more than one child subgraphs in a component graph, then number of loops ($L_{Gc,j}$) and independent loops ($L_{c,j}$) present in each child subgraph are determined with the following equations (eqns. (6.27) and (6.28)) by finding the number of vertices and edges present in each child subgraph.

$$L_{Gc,j} = 2S + E_j - V_j \qquad \forall \ \ j \in J, \text{ such that } g_j \in G_c \tag{6.27}$$

where E_j and V_j are edge set and vertex set of j^{th} subgraph belonging to the child subgraph set G_c and independent loop $L_{c,j}$ of j^{th} child subgraph is

$$L_{c,j} = L_{Gc,j} - S. \tag{6 28}$$

There are two child subgraphs (g_2, and g_3) for the component graph shown in Fig. 6.3(b). Vertices and edges present in child subgraphs determined by vertex fusion methodology are given in Table 6.3. There are 8 vertices, and 10 edges in child subgraph g_2, and 4 vertices and 4 edges in second child subgraph g_3. Therefore, number of loops ($L_{Gc,1}$, and L_{Gc2}) present in g_2, and g_3 subgraphs respectively are identified in accordance with eqn. (6.28), which are given below :

$L_{Gc,2} = 2 \times 1 + 10 - 8 = 4$

and

$L_{Gc,3} = 2 \times 1 + 4 - 4 = 2$

Independent loops ($L_{c,2}$ and $L_{c,3}$)) present in child subgraphs (g_2, and g_3) are identified by eqn 6.28, which is given below :

$L_{c,1} = 4 - 1 = 3$

and

$L_{c,2} = 2 - 1 = 1$

Loops in each child subgraph are identified in the similar way as that of the identification of loops of the parent subgraph

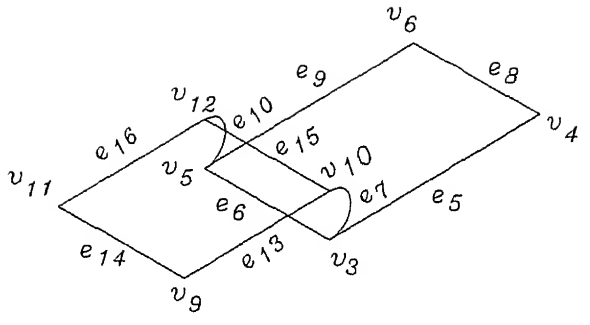
A child subgraph g_2 of the component graph G (Fig. 6 3(b)), and its EA matrix along with degree of edge ($d(e_m)$) and loop degree ($l(e_m)$) for edges present in g_2 are given in Fig. 6 7(a) Similarly, child subgraph g_3 of the component graph G (Fig 6.3(b)), and its EA matrix along with degree of edge ($d(e_m)$) and loop degree ($l(e_m)$) for edges present in g_3 are given in Fig. 6.8(a). The output of the the system is given in Figs 6.11(b) and 6.12(b) for g_2 and g_3 respectively. The number of loops identified are three ($l_{g2,1}$, $l_{g2,2}$, $l_{g2,3}$) in case of g_2 subgraph, and one ($l_{g3,1}$) in case of g_3 child subgraph.

6.3.4 Identification of child subgraph loop present in the parent subgraph loop

A child subgraph loop is said to be located in a loop of the parent subgraph, if the plane equation of the loops are the same In a child subgraph if there are more than one loops, then at least one loop lies inside a loop of parent subgraph, and the other loops may be either inside a loop of the parent subgraph or outside. On the other hand if there is only one loop in a child subgraph, then it will lie inside one of the parent subgraph loops

Implementation

The plane equation of loops are determined while finding independent loops in parent and child subgraphs. Each child subgraph loop is processed to determine the loop of the parent subgraph in which it contained by comparing the plane equations. Once the loop of a child subgraph located in a plane is determined, then the process is continued for another child subgraph loop till all the loops in the child subgraph are over. The flowchart for the same is shown in Fig. 6 9, and the output of the system is given in Fig. 6.7(b) and Fig. 6.8(b) for child subgraphs g_2 and g_3 respectively of the component graph given in Fig. 6.3(b).



	5	6	7	8	9	10	13	14	15	16	$d(e_m)$	$l(e_m)$
5	–	1	1	1							3	1
6	1	–	1		1	1					4	2
7	1	1	–				1		1		4	2
8	1			–	1						2	1
9		1		1	–	1					3	1
10		1			1	–			1	1	4	2
13			1				–	1	1		3	1
14							1	–		1	2	1
15			1			1	1		–	1	4	2
16						1		1	1	–	3	1

(a)

Loop in a child subgraph $l_{g2,1}$ e_5 e_6 e_8 e_9

Loop in a child subgraph $l_{g2,2}$ e_6 e_7 e_{10} e_{15}

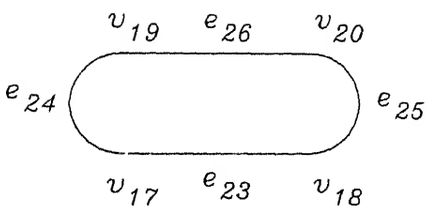
Loop in a child subgraph $l_{g2,3}$ e_{13} e_{14} e_{15} e_{16}

Common edges e_6 e_{15}

Child subgraph loop located in the parent subgraph loop $l_{g2,1}$ in $l_{g1,1}$

(b)

Figure 6.7. (a) Child subgraph g_2 of the component graph and EA matrix along with degree of edge and loop, and (b) Edges present in different loops, and child subgraph loop located in the parent subgraph.



	23	24	25	26	$d(e_m)$	$l(e_m)$
23	–	1	1		2	1
24	1	–		1	2	1
25	1		–	1	2	1
26		1	1	–	2	1

(a)

Loop in a child subgraph $l_{g3,1}$ e_{23} e_{24} e_{25} e_{26}

Child subgraph loop located in the parent subgraph loop $l_{g3,1}$ in $l_{g1,4}$

(b)

Figure 6.8: (a) Child subgraph g_3 of the component graph and EA matrix along with degree of edge and loop, and (b) Edges present in different loops, and child subgraph loop located in the parent subgraph.

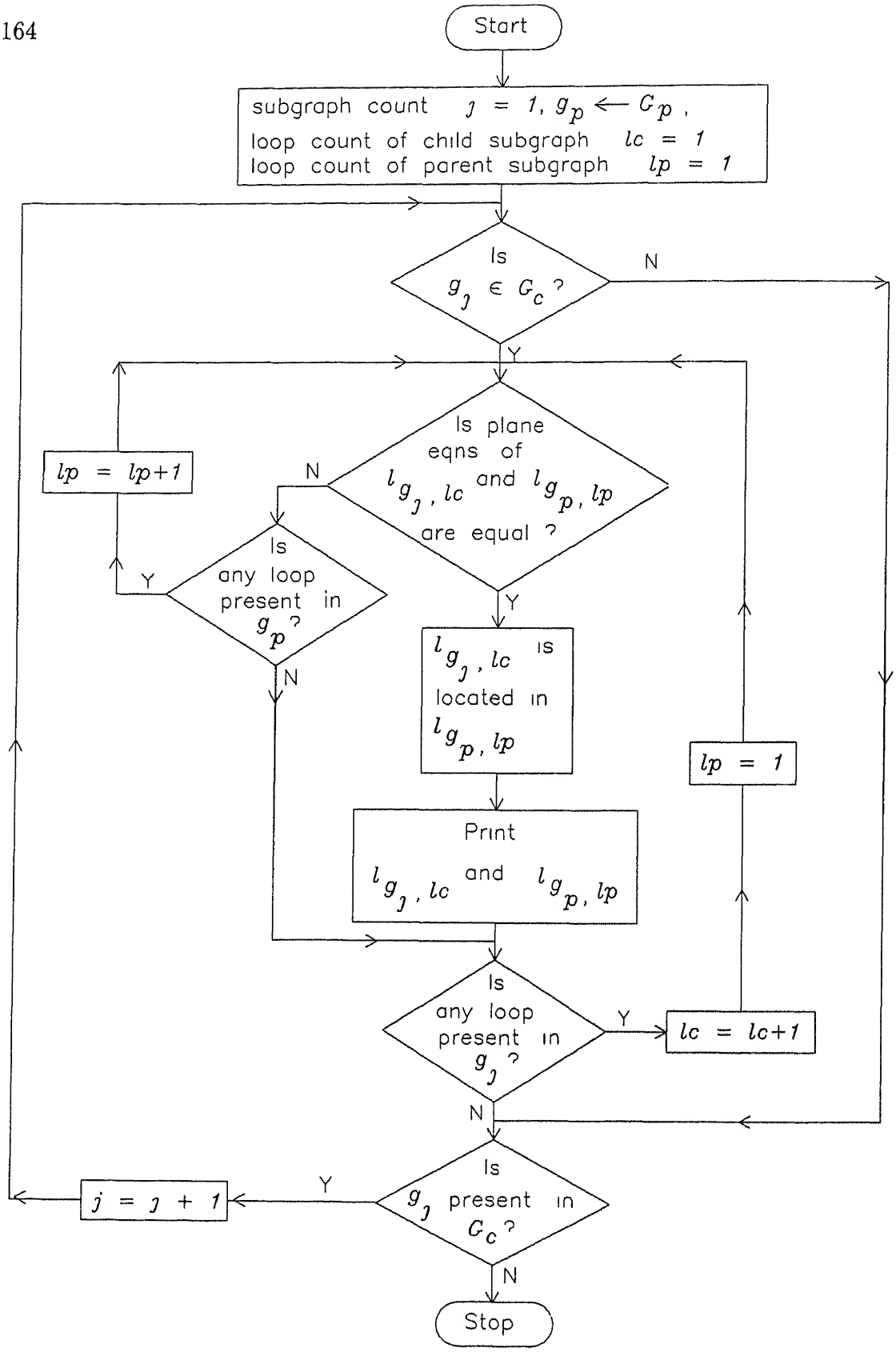


Figure 6.9: Flowchart for identifying child subgraph loop located in the parent subgraph loop.

Bend plane	$bp_{g_1,1}$	$l_{g_1,2}$
Bend plane	$bp_{g_2,1}$	$l_{g_2,2}$

Table 6 5 Curved planes present in the component graph

6.4 Recognition of a curved plane in the component

Bend allowance is calculated for curved surfaces of a component to get its flat pattern development. In wireframe model, a curved plane is represented by curved edges, i.e., arcs. A loop having an edge as an arc is selected A point (other than the start and end vertices) lying on the segment of the arc is determined. If the point satisfies the plane equation (eqn. (6 25)) of the loop under consideration, then the plane is linear, otherwise it is a curved plane

Implementation

Flowchart for identifying a curved plane in the component from its graph is given in Fig. 6.10 It should be noted that a subgraph is considered for finding the curved surface only if it has more than one loop. Output of the above methodology for the component graph shown in Fig. 6.3(b) is given in Table 6.5.

6.5 Identification of adjacent planes

For recognizing forming features in a component, the adjacency relationship between planes is required. It should be noted that loops identified in the parent subgraph of a component graph represent planes or forming features, or both in a component as discussed earlier Also, in case of child subgraphs of a component graph, if the number of loops identified are more than one, then at least one of the loops lies inside the plane of the parent subgraph, and other loops which are not present in the parent subgraph loop represent different planes in the component. On the other hand, if there is only one loop in a child subgraph, then it is located inside the plane of the parent subgraph A plane is said to be adjacent to the other plane, if there is at least one common edge between the two loops in a subgraph. The flowchart for identifying an adjacency plane relationship (APR) matrix for parent and child subgraphs is given in Fig. 6.11.

Adjacency plane relationship (APR) matrices of parent and child subgraphs of the component shown in Fig 6 3(a) obtained with the above method are given in Fig 6.12.

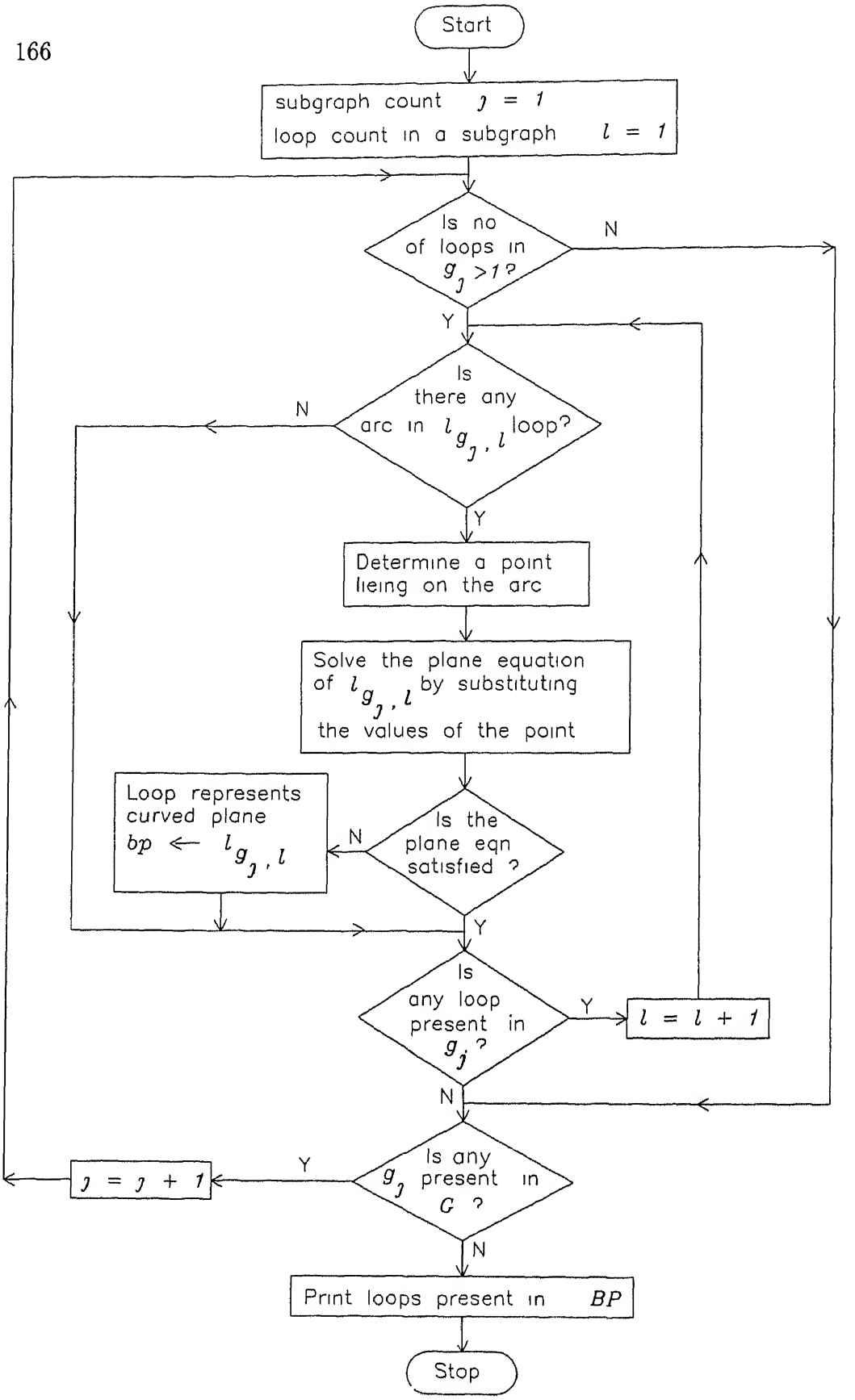


Figure 6 10: Flowchart for identifying curved plane in a component graph.

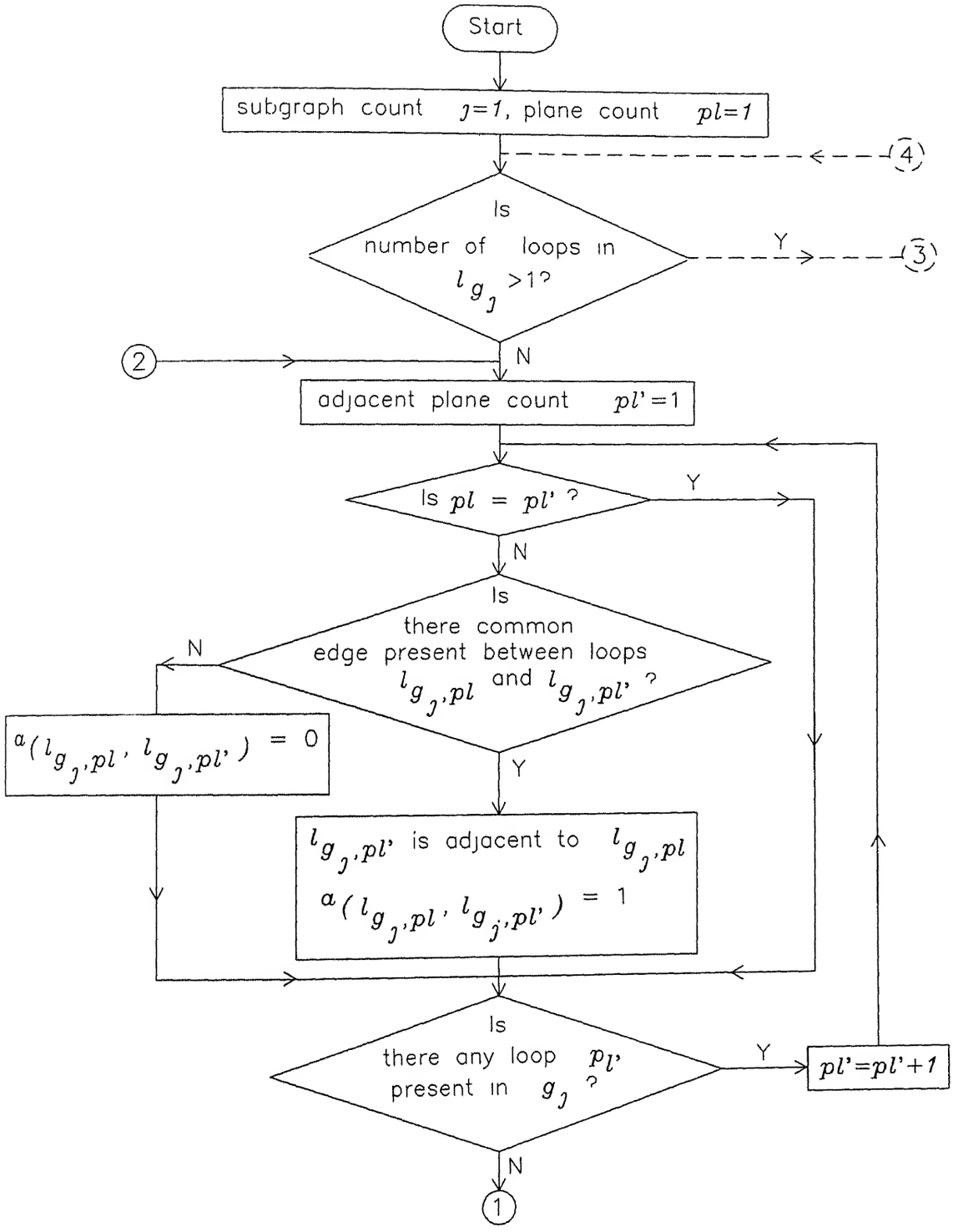


Figure 6.11: Flowchart for determining adjacency plane relationship (APR) matrix for a component graph.

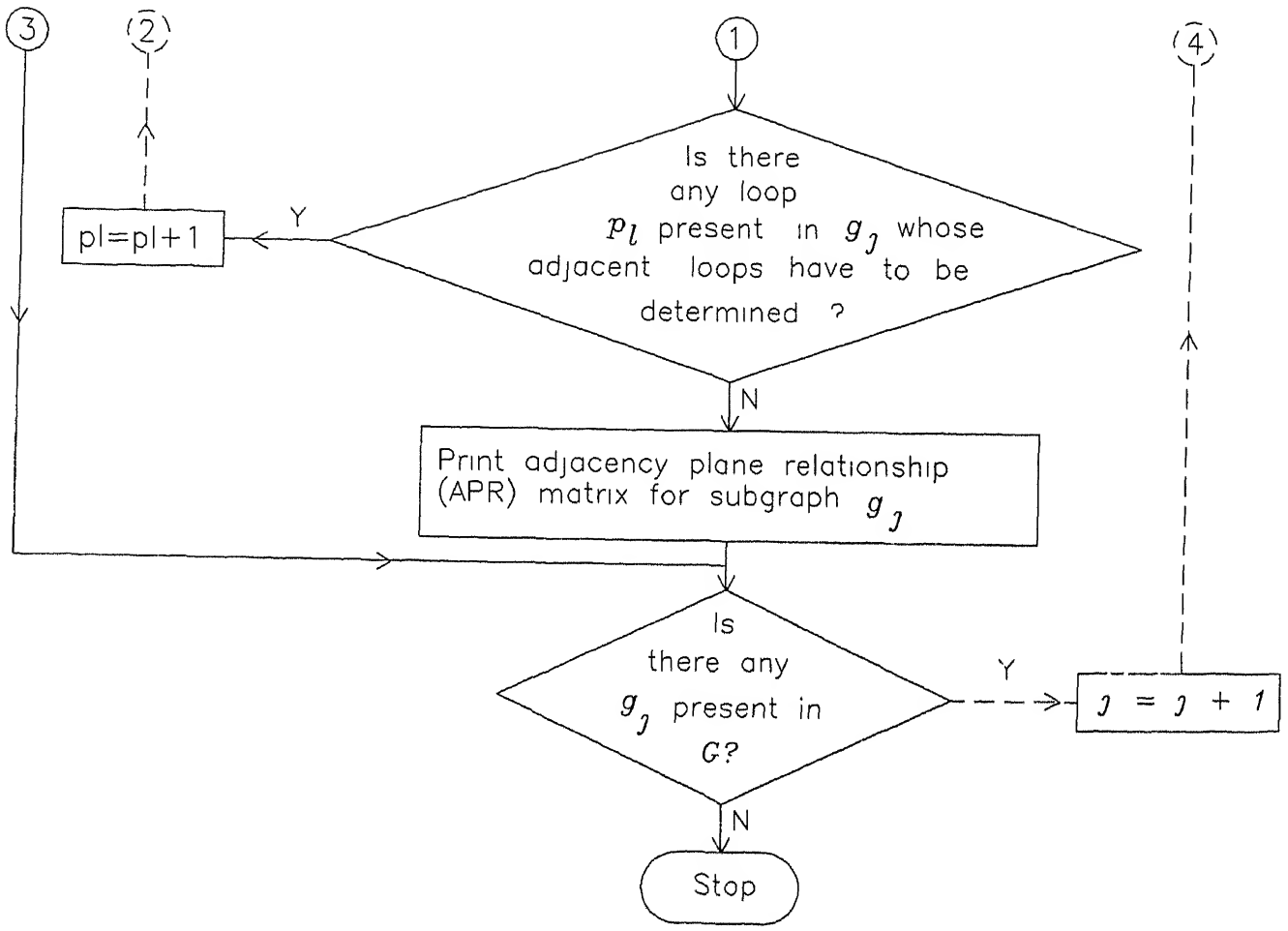


Figure 6.11: Flowchart for determining adjacency plane relationship (APR) matrix for a component graph (continued).

lg_1, pl' / lg_1, pl	$lg_1, 1$	$lg_1, 2$	$lg_1, 3$	$lg_1, 4$	$d(lg_1, pl)$
$lg_1, 1$	–	1			1
$lg_1, 2$	1	–	1		2
$lg_1, 3$		1	–	1	2
$lg_1, 4$			1	–	1

(a)

lg_2, pl' / lg_2, pl	$lg_2, 1$	$lg_2, 2$	$lg_2, 3$	$d(lg_2, pl)$
$lg_2, 1$	–	1		1
$lg_2, 2$	1	–	1	2
$lg_2, 3$		1	–	1

(b)

Figure 6.12: (a) APR matrix of the parent subgraph, and (b) APR matrix of the child subgraph.

In the APR matrix, the value of the cell $(a_{l_{g_j,pl},l_{g_j,pl'}})$ is either '1' or '0' ('0' is represented by an empty cell). If the cell value is '1', then the loop $l_{g_j,pl'}$ is adjacent to $l_{g_j,pl}$ and if it is '0', then the loop $l_{g_j,pl'}$ is not adjacent to $l_{g_j,pl}$, i.e.,

$$\text{If} \quad a_{l_{g_j,pl},l_{g_j,pl'}} = \begin{cases} 1 & \text{if there is at least one common edge between two loops} \\ & \text{in the subgraph } l_{g_j,pl} \text{ and } l_{g_j,pl'} \\ 0 & \text{otherwise.} \end{cases} \tag{6.29}$$

6.6 Cross-bend feature extraction

Forming features other than the features (cross-bend features in Fig. 6.13) located in different adjacent planes in a subgraph of a component graph can be identified directly from the subgraphs recognized. Features present in a bend feature zone and its adjacent planes are termed as cross-bend features.

A cross-bend feature can be present both in parent and child subgraphs Cross-bend features present along the curvature bend and sharp bend zone in a component are shown in Fig. 6 13(a) and the component graph is given in Fig. 6.13(b). The methodology used to identify these type of features is explained below :

Step 1. Determine the number of common edges present between two adjacent planes in a subgraph

Step 2. If the adjacent planes in a subgraph have more than one common edge, then it indicates that the adjacent planes under consideration of the subgraph have cross-bend features

In Fig. 6 13(b), two common edges e_7 and e_{10} are present between flat plane $fp_{g1,1}$ and curved plane $bp_{g1,1}$ Therefore, the subgraph is processed to delete the common edges present in it. After deletion of common edges from the subgraph, it is given in Fig. 6.14(a). Common edges in a subgraph are determined as explained in Section 6.3.1 of identification of loops

Step 3. Different sub-subgraphs present in a subgraph under consideration are determined by the vertex fusion methodology as explained in the Section 6.2.

Step 4. The sub-subgraph identified in Step 3 form a loop. The loops located inside another loop of a sub-subgraph are identified as cross-bend features.

Implementation

During implementation, the rows and columns representing vertices of common edges are not deleted from the vertex adjacency (VA) matrix of the graph, but they are discarded from the VA matrix by assigning a flag, and vertex fusion methodology is carried out to identify different sub-subgraphs. The identified cross-bend features of the component graph shown in Fig. 6.13(b) along with vertices and edges present in each cross-bend feature are given in Fig. 6.14(b).

6.7 Recognition of forming features

In the present work, combination of geometry reasoning and pattern recognition process is employed as a strategy for recognizing the forming features. The recognition process is of

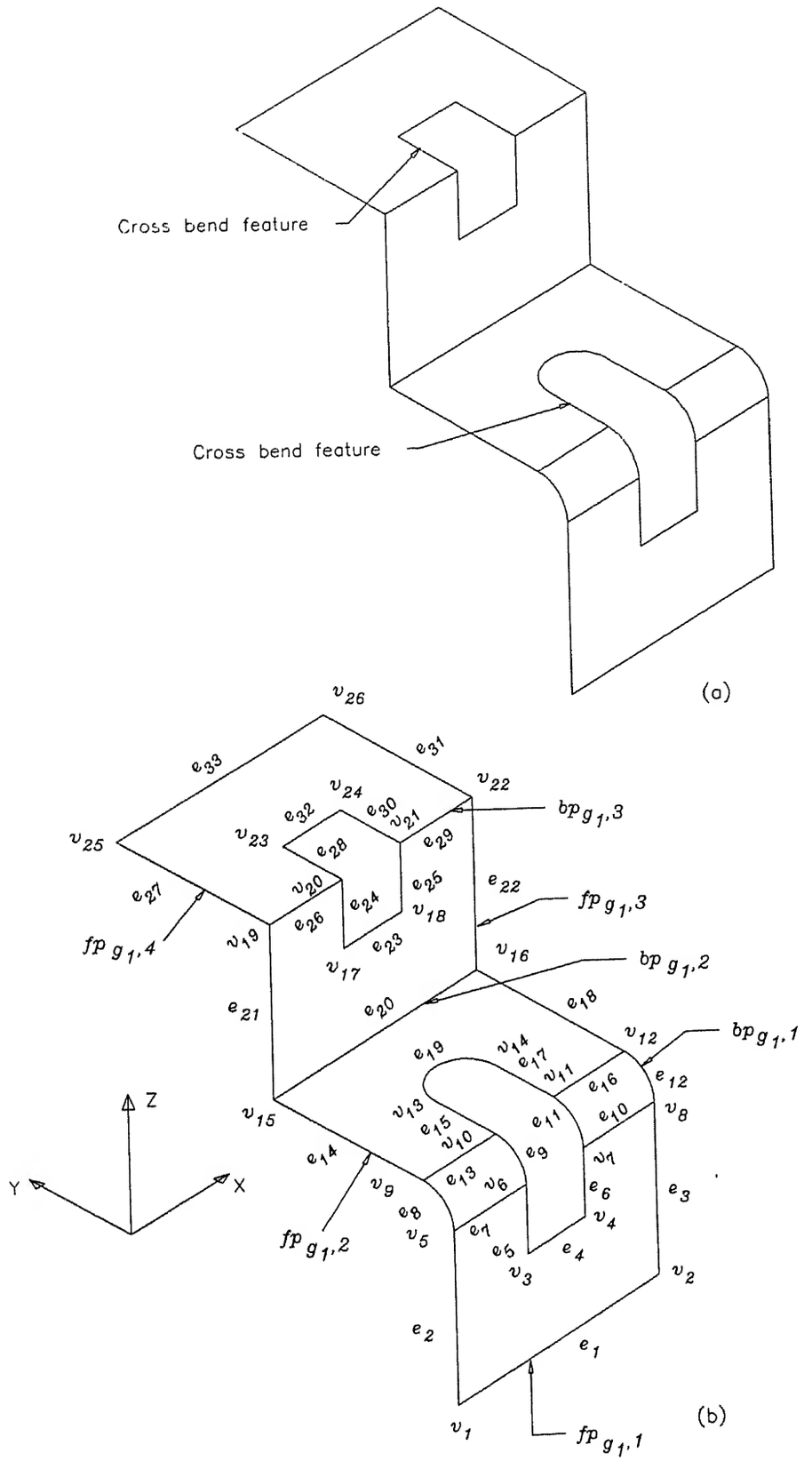
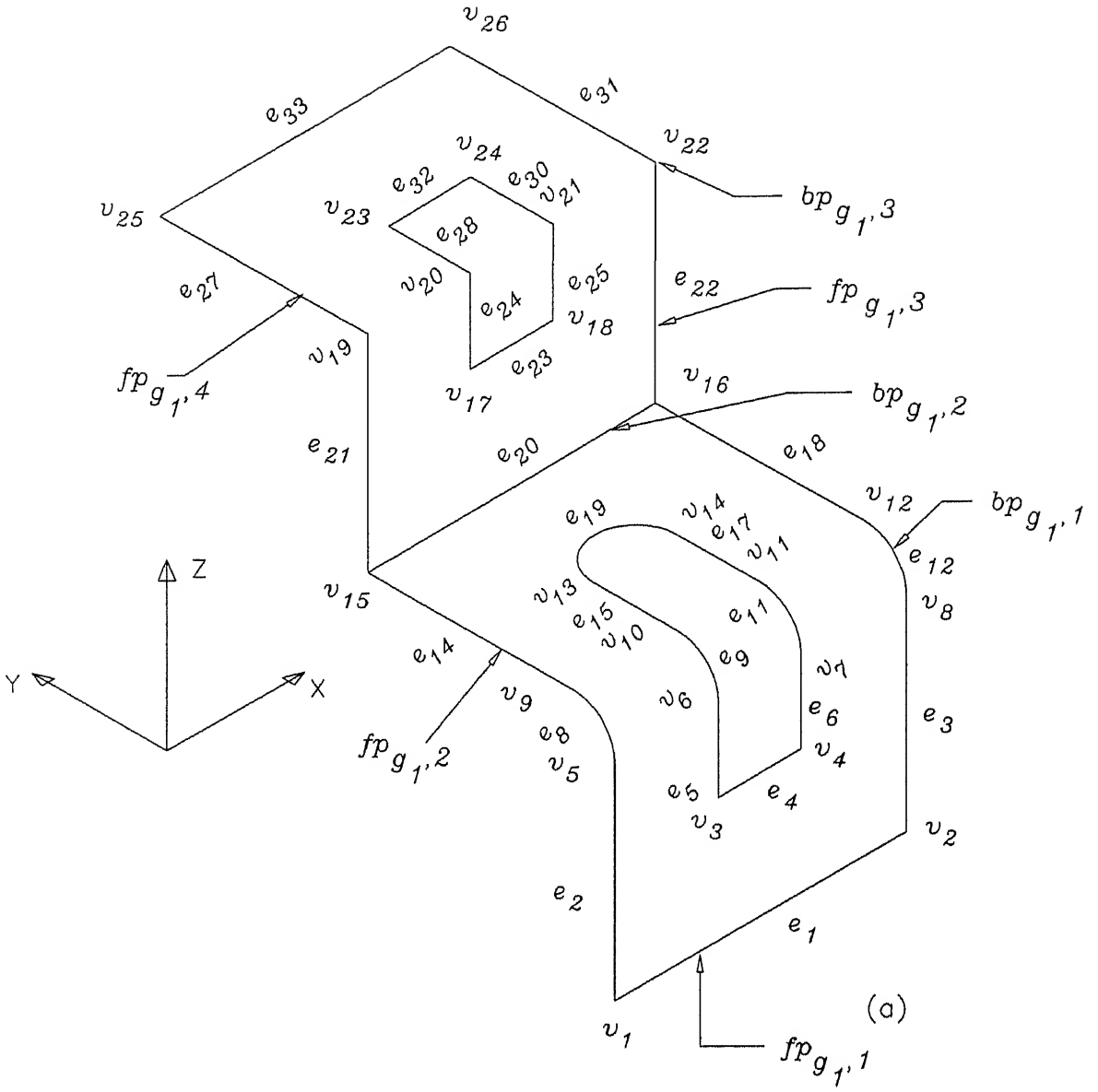


Figure 6.13: (a) A component having cross-bend features, and (b) Component graph.



Cross bend feature	$cbf_{g_{1},1}$	v_3 v_4 v_{10} v_{11} v_{13} v_{14}
Cross bend feature	$cbf_{g_{1},2}$	v_{17} v_{18} v_{20} v_{21} v_{23} v_{24}
Cross bend feature	$cbf_{g_{1},1}$	e_4 e_5 e_6 e_9 e_{11} e_{15} e_{17} e_{19}
Cross bend feature	$cbf_{g_{1},2}$	e_{23} e_{24} e_{25} e_{28} e_{30} e_{32}

(b)

Figure 6.14: (a) The component graph after removing common edges between planes having cross-bend features, and (b) Vertices and edges present in cross-bend features.

two levels :

(i) In the *first level*, entities belonging to an entity group are identified. From this entity group (which represents subgraphs of a component graph) parent and child subgraphs are identified. In each subgraph, loops are determined and their type (i.e., curved or linear type) are recognized along with their geometrical parameters. Adjacency plane relationship (APR) matrix for all the subgraphs is generated and subgraphs are further processed to extract cross bend features. Thus the first level of feature recognition is general in nature.

(ii) In the *second level*, which is domain specific, the preprocessed information obtained from the first level is utilized to recognize forming features with the help of pattern recognition methodology In pattern recognition process, features are identified by matching (i.e., by comparing the properties) the patterns of features developed with the patterns extracted from the component If the two feature graphs are isomorphic, then the pattern is said to be matched and the feature is present in the component graph, otherwise not. Two graphs are said to be isomorphic not just based on the adjacent relationships of nodes, but when all the information present in the nodes and edges of the two feature graphs are same.

Let $FG_1(V_1, E_1)$ and $FG_2(V_2, E_2)$ be two feature graphs and $\psi : FG_1 \Longrightarrow FG_2$, where ψ be isomorphism, which maps $FG_1(V_1, E_1)$ onto $FG_2(V_2, E_2)$ Isomorphism ψ is defined such that

1. $\forall\, v_i \in V_1$, $\exists\, v_i' \in V_2$, such that $\psi(v_i) = v_i'$

2 $\forall\, e_m \in E_1$, $\exists\, e_k' \in E_2$, such that $e_k' = (\psi(v_i), \psi(v_i'))$.

Various forming features and their graphical representation along with geometrical and topological information required to represent them are discussed in the following section.

6.7.1 Characteristics of forming features

A forming feature *is a set of geometric entities consisting of edges which are spatially connected such that they can be manufactured by forming operations.* Forming features are found between planes, inside the planes or at the sides of a component.

Forming features are broadly classified into three categories (Fig. 2.8) viz., parent subgraph features, child subgraph features and parent–child subgraph features Classification system of forming features is based on the classification system of forming operations (Fig.2.7), and is discussed in Section 2.5.2. Features considered for recognition in the present work are bending, curling, contour roll, flanging, hemming, joggling, and louvering. The definitions of the above operations and figures are given in Appendix B. Graphical representation of features, and geometrical and topological information associated with

them are given in Figs 6.15 to 6 17. The principle of isomorphism of graphs is used to recognize features extracted from the component graph with the feature graphs developed, and geometrical and topological information associated with the feature graphs Graphical representation, and the geometrical and topological information associated with different types of forming features are explained below ·

Bending feature characteristics

Bending operation is used to form *linear sections from the flat sheets* Therefore, bending features are present between two planar planes of the parent subgraph of a component graph. The bend feature represents a loop or a path. As shown in Fig. 6 15(a(i)), graph of the *curvature bend feature* (i.e., mean radius of bend is not zero) is a loop, and consists of four nodes and four edges. If the bend is of *sharp type* (Fig. 6.15(a(ii)), i.e. mean radius of bend is zero, then the graph represents a path, and is made of two nodes and one edge Thus, the bending features are characterized as :

1 Bending feature graph is present in the parent subgraph.

2 Bending feature is present between two planar planes, such that they are adjacent to the bend feature.

3. Curvature bend feature is represented by a single curved plane, and sharp bend feature is represented by an edge, which is the intersection of two adjacent planar planes.

4. The graph of a curvature bend is a loop, having four vertices and four edges, such that the two vertices located in the adjacent planar planes of curved plane are connected by arc type of entities (vertices v_1 and v_3 in Fig. 6.15(a(i))).

5. Graph of a sharp bend feature is a path, and it consists of two vertices and one edge.

It should be noted that during pattern recognition, the type of bend feature graph selected is based upon the type of bend (i.e. curvature bend or sharp bend).

Curling feature characteristics

Curling process is used for *forming an edge of circular cross section along the length or width of a component.* Hence, curling feature graph represents a loop and it is the end plane of a component (Fig. 6.15(b)). The curling feature has following characteristics :

1. Curling feature graph is present in the parent subgraph.

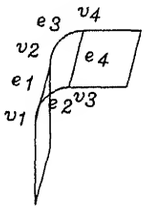
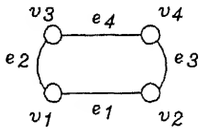
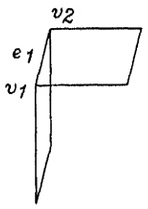
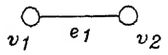
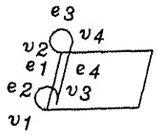
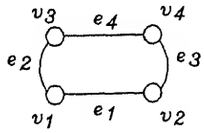
Sl no	Feature name	Feature representation	Graphical representation	Geometrical and topological information
a(i)	Curvature bend			Feature graph is present in the parent subgraph Curved plane is present between two flat planes Feature graph consists of 4 vertices and 4 edges in case of curvature bend, and 2 vertices and 1 edge in case of a sharp bend In curvature bend, arc type of entities connects the flat planes, and in sharp bend, the flat planes intersect each other
a(ii)	Sharp bend			
b	Curling			Feature is present in the parent subgraph Curved plane is present at the sides of the component, Feature graph consists of 4 vertices and 4 edges Curved plane edges connecting the planar planes are of arc type entities

Figure 6.15: Graphical representation, and geometrical and topological information of forming features present in a single plane of the parent subgraph.

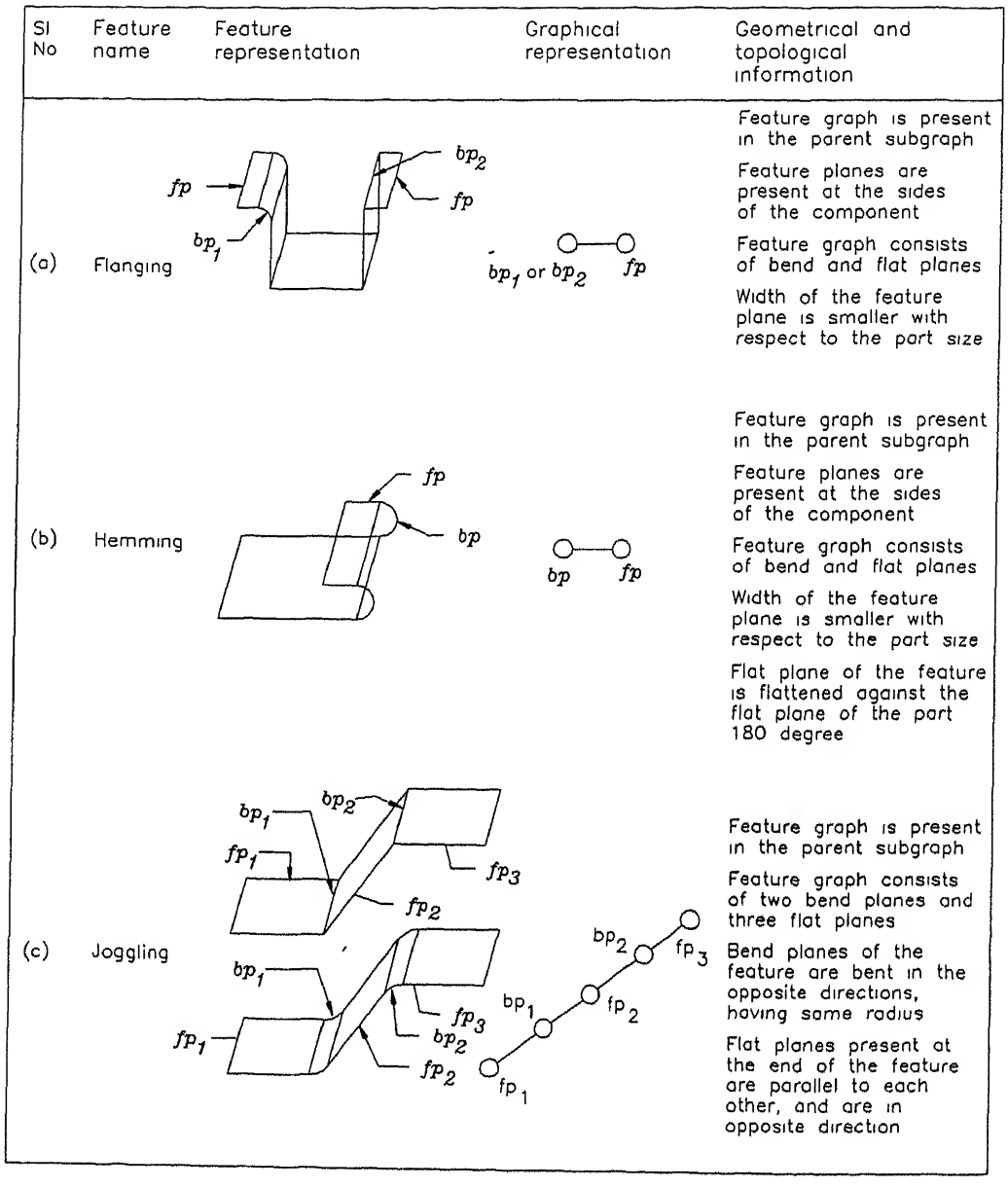


Figure 6.16: Graphical representation, and geometrical and topological information of forming features present in a few multi-planes of the parent subgraph.

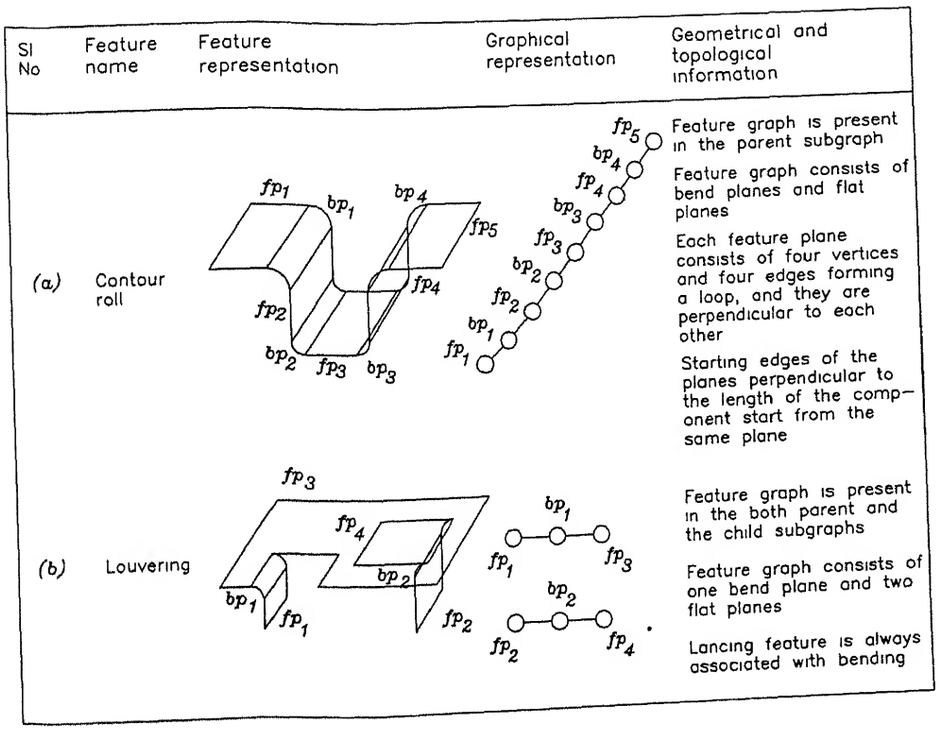


Figure 6.17: Graphical representation, and geometrical and topological information of forming features present in all planes of the parent subgraph, and in a few planes of parent and child subgraphs

2. Curling feature is the end plane of a component. It is connected to a flat plane of the component on one side

3 It is represented by a single curved plane

4. The graph of the feature is a loop, and has four vertices and four edges such that, the edges connecting the vertices of one degree and the vertices located in the adjacent planar plane are arc type of entities.

Flanging feature characteristics

Flanging is a *process of forming an edge and is similar to bending, except that the metal bent down is shorter when compared to the overall part size.* Thus a flange (Fig 6.16(a)) is the end plane of the component The adjacent plane of the flange may be curvature bend feature or an edge representing sharp bend feature as shown in Fig. 6 16(a). Flanging operation is done in a flanging die such that the bend feature and the flange (i.e. short plane) associated with it are done simultaneously. Therefore, flanging feature is classified as multi-plane feature, and its characteristics are summarized below :

1. Flanging feature is present in the parent subgraph

2. Flange is present at the end plane of the component.

3. Flanging feature is a multi-plane feature.

4. The graph is represented by bend plane and flat plane.

Hemming feature characteristics

Hemming is an operation in which *flanges are flattened against the workpiece by 180 degree bends to make a finished or reinforced edge.* Hence, the hemming feature is present in the parent subgraph, and it is multi-plane feature (fp and bp together in Fig. 6.16(b)). Its representation is similar to that of the flanging feature, except that the end plane (fp) adjacent to the bend plane (bp) will be either parallel to another adjacent plane of bp by 180 degrees or the flat end plane is bent more than 180 degrees, as in the case of tear drop hems (Appendix B).

Joggling feature characteristics

Joggling operation is an operation, in which *offset of flat planes (fp_1 and fp_2) is obtained by two parallel bends in opposite directions at the same angle.* Thus, joggling feature (Fig. 6.16(c)) is present in a parent subgraph, and it is a multi-plane feature. The graphical representation of the feature with respect to planes is given in Fig. 6.16. The geometrical and topological characteristics of a joggling feature are :

1. Joggling feature is present in the parent subgraph

2 Joggling feature is a multi-plane feature

3 Joggling feature graph consists of three flat planes and two bend planes in it, and they are placed in the feature graph as shown in Fig 6 16

4. The end planes of the feature are planar planes, and they are parallel to each other in the opposite directions.

Contour roll feature characteristics

In contour roll process, *the desired shape for long components of uniform cross section is produced by feeding the raw material longitudinally through a series of contoured rolls* Therefore, the entire component forms the contour roll feature (Fig. 6.17(a)), provided the component has uniform cross section. The contour roll feature is characterized as given below :

1. Contour roll feature is present in the parent subgraph

2. Each loop (plane) in the component graph will have only four vertices and four edges, since the cross section of the component is uniform in the longitudinal direction.

3. Lengths of the planes are same, angle between adjacent edges of a plane is 90 degrees, and starting edges of the planes perpendicular to the length of the component start from the same plane.

Louvering feature characteristics

Louvering is a process in which *both lancing and bending operations are done.* Thus, louvering feature (Fig. 6.17(b)) has bend plane feature, and the planes adjacent to this bend feature are flat planes. If the feature is present in the child subgraph (Fig. 6.17(b)), then one of the loops (fp_2) adjacent to the bend feature (bp_2) represents flat plane, and other adjacent loop located in the plane of the parent subgraph loop represents lancing feature, which is a 2–D plane feature. In case of its presence in parent subgraph, then the adjacent planes of the curved plane are flat planes The louvering feature is characterized as given below :

1 Louvering feature is a parent–child subgraph feature. It may be present in both parent as well as child subgraphs.

2. Louvering feature is a multi-plane feature, having bend plane between two flat planes as shown in Fig.6. 22(b).

3. Lancing feature and bend feature are associated with louvering feature.

6.7.2 Feature recognition

Features are recognized by deriving a set of principles to relate them geometrically and topologically. As in case of recognition of shearing features, all forming features are identified for a component. This is due to the fact that identification of various operations by means of which a component can be manufactured helps in the development of alternative process plans and in turn, an optimized process plans from the alternative process plan can be selected by taking various constraints into account. Initially bend features are identified in a subgraph. Then, end planes and its type (i e curved or flat) are determined in a subgraph. Afterwards, features are recognized by matching test graph features with the pattern graph represented earlier for bending and curling. Matching is done by both heuristic process and isomorphic equality process. Also, by this process other forming features are recognized after combining adjacent planes The heuristic and isomorphic equality processes are explained in the following sections

Heuristic process

In Heuristic process, a pattern graph is rejected if it is not isomorphic with the test graph under consideration. Thus heuristic procedure helps in rejection of a pattern This procedure is in accordance with the definition of isomorphism of graphs (Deo, 1990). The definition of isomorphism states that two graphs may be isomorphic if the following conditions are satisfied, since the conditions are essential for graphs to be isomorphic, but they are not sufficient. The conditions are :

1. Number of vertices in a test graph should be same as that of a pattern graph
2. Number of edges in both the test graph and pattern graph has to be same.
3. Equal number of vertices with a given degree should be same in test and pattern graphs.

If pattern graph and test graph satisfy the above conditions, then the test graph is further considered for its isomorphic equality.

Isomorphic equality

Heuristic method is applied for checking the isomorphic equality of two graphs, which depends on the classification system of features. In this method, adjacency matrices along with geometrical and topological information of test feature graph (TFG) and pattern feature graph (PFG) are compared . If the adjacency matrices of feature graphs (TFG and PFG) are same, and information associated with them given in Fig 6.15, Fig. 6.16,

and Fig 6.17 for each feature graph are also same, then the test feature graph is said to be isomorphic with the pattern feature graph under consideration. Determination of adjacency matrices for vertices, edges and planes (i.e , VA, EA and APR matrices) have been explained earlier in the thesis.

Recognition of parent subgraph features present in a single plane

In isomorphic equality of forming features present in a single plane of parent subgraph, the feature graphs of pattern and test are said to be same, if the vertex and edge adjacency (VA and EA respectively) matrices of pattern feature graph (PFG) and test feature graph (TFG) are same Also geometrical and topological information associated with the PFG (Fig 6 15) should satisfy by the TFG

Recognition of parent subgraph features present in a few multi-planes

Adjacency plane relationship (APR) matrix of a bending feature is determined after bending features present in the parent subgraph are identified as given in the previous section. Further based on the the geometrical and topological information associated with the PFG (Fig. 6.16), TFG are said to be isomorphic with the appropriate PFG.

Recognition of parent subgraph features present in all planes

In this type of isomorphic equality checking, the TFG is tested for only the cross section of the component. If it satisfies the geometrical and topological information given in Fig. 6.17, the TFG is said to be isomorphic with PFG

Recognition of parent–child subgraph features

After the development of flat pattern, the component is processed to identify the lancing feature. If the lancing feature is associated with the bend feature (adjacent plane information is determined from APR matrix), then the isomorphic equality process is carried out. The combination of lancing and bending features of TFG is said to be isomorphic with louvering PFG (Fig. 6 17(b)), if geometrical information and topological information associated with PFG is satisfied. In the following section flat pattern development methodology for a 3–D component by taking thinning effect into consideration is given.

6.8 Flat pattern development for a component

Flat pattern development for a component is defined as the resulting shape of the development of surfaces which refer to the process of unfolding or unrolling a surface onto a plane (Pickup, et all., 1972). Flat pattern of a component is required to cut the exact shape and

size of the blank to manufacture the component, and for nesting to utilize the raw material optimally

Nee and Chong (1988), and Loh et al , (1992) have used flat pattern development for sheet metal parts of heating, ventilating and air-conditioning ducts using slab method (Sachs and Voegeli, 1966) for calculating the bend allowance Nnaji et al , (1993) have also used slab method for finding bend allowance to develop flat pattern for components created in a solid model. Also, presently in many sheet metal industries, a flat pattern is developed manually using the slab method for calculating bend allowance Prasad and Somasundaram (1993) have developed a mathematical model for calculating bend allowance in automated sheet metal bending by taking thinning effect into consideration. The bend allowance parameter is calculated by assuming plane-strain deformation and a rigid-plastic raw material.

Before further discussion on the flat pattern development, terminologies used in bending are given in Fig. 6 18 (Sachs and Voegeli, 1966, Lascoe, 1988, and Metals handbook, Vol 4, 1979), for easy understanding of calculating the bend allowance.

Bending terminologies and definitions

Bend radius (R_{in}) : Bend radius is defined as the part radius on the inside of the bend.

Bend angle (A) : Bend angle is the actual angle between two flat portions or, if these are missing, then it is given by the angle between the tangents to the bend curvature at both ends.

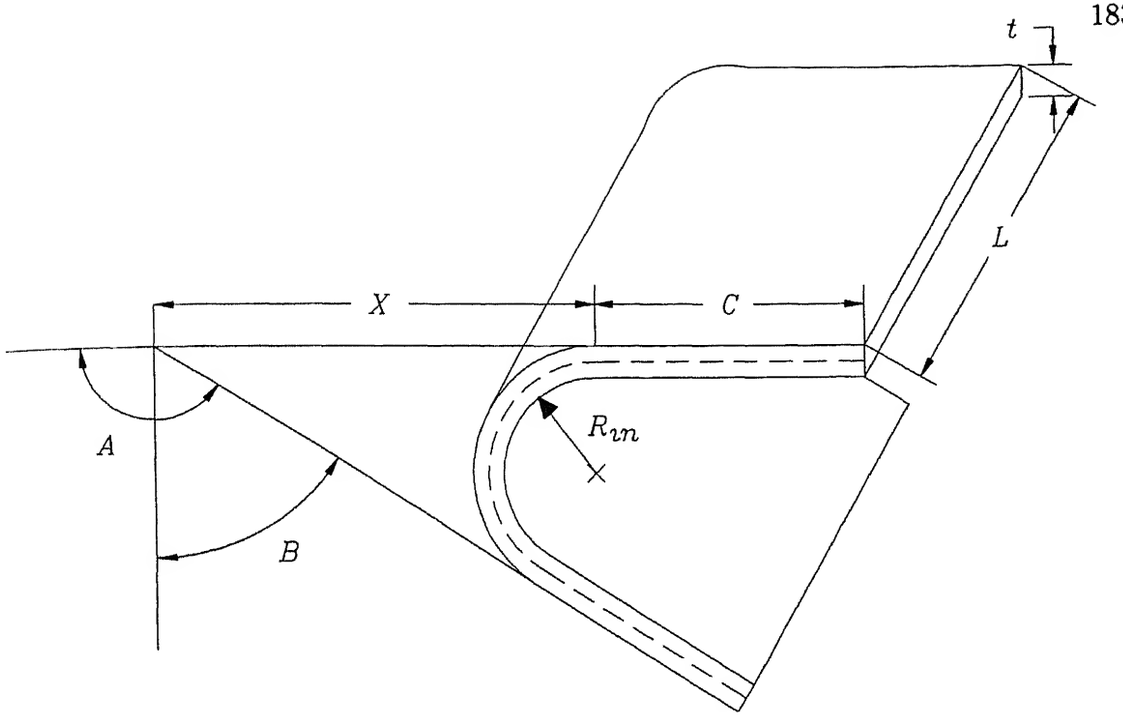
Bend length (L) : It is the length of the bend.

Bevel angle (B) : Bevel angle is the difference between the bend angle and 90 degrees.

Set back (X) : Distance between outer mold line to the bend line of the part is called set back.

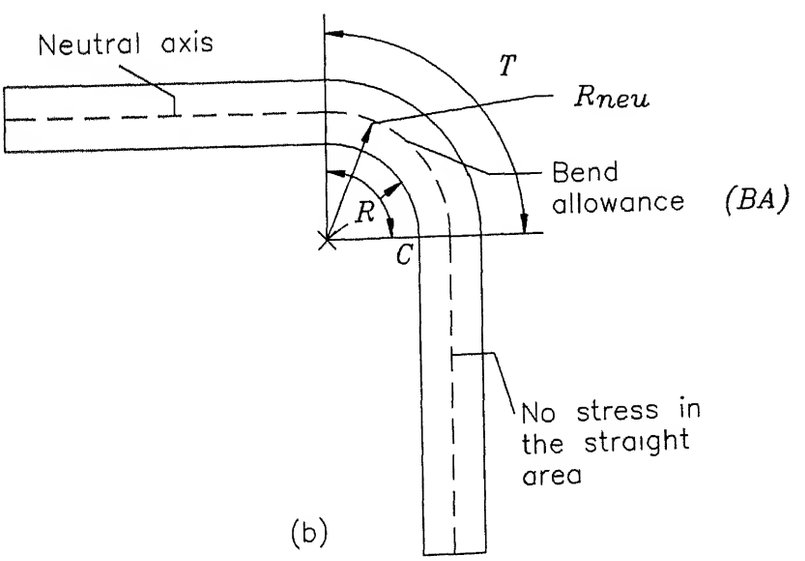
Neutral axis (R_{neu}) : In neutral axis, the stress is zero, and it is the axis in which the length of the bend is retained.

Bend allowance (BA) : The developed or initial length of the center line of the bent arc is termed as bend allowance.



A	Bending angle	L	Length of bend
B	Bevel angle	R_{in}	Bend radius
C	Leg of flange or width of web	t	Thickness of material
		X	Set back

(a)



(b)

Figure 6.18: (a) Bending terminologies, and (b) Bending stresses.

6.8.1 Calculation of bend allowance in a component and implementation

In the slab method used for calculating bend allowance, the thickness change occurring in the bend zone is neglected. Therefore, the radius of curvature of the neutral axis would be equal to the sum of bend radius and half of the metal thickness, and the bend allowance is given by

$$BA = (R_{in} + t/2) * (\pi * A)/180. \quad (6\ 30)$$

Thus in slab method only geometry of the bend is taken into consideration, neglecting the material properties.

In case of thinning effect consideration (Prasad and Somasundaram, 1993), strain hardening of the material is considered along with geometrical properties of the bend for calculating the bend allowance In this research of feature recognition of forming operations, a flat pattern development of a sheet metal component is done based on the mathematical model developed by Prasad and Somasundaram (1993) for calculating bend allowance. This is essential as we know that a curved plane changes its length slightly along the center line and will be larger than its length in the blank, when the bend allowance is calculated This is due to the fact that bending results in a decrease of metal thickness in the bend zone. It should be noted that the bend allowance becomes smaller with decreasing bend radius. Any inaccuracy in the calculation of bend allowance may result in increasing the inaccuracy in the final flat pattern development of the component. This will in turn lead to the incorrect blank size, and hence in the location of bend lines, slots, holes, etc., as well as incorrect length in cross bend features

In the model proposed by Prasad and Somasundaram (1993), the bend allowance is calculated along the fiber, free of strain, which is called the unstretched fiber, and the stress free fiber is called neutral fiber.

The nomenclature used in the development of necessary equations for bend allowance are given below in accordance with the Fig. 6.19 and Fig. 6.20.

BA bend allowance.
e_{in} surface strain at innermost fiber.
e_{out} surface strain at outermost fiber.
k strength coefficient of material.
n strain hardening exponent.
R_{in} inner radius of bend.
R_m radius of mean fiber.
R_{neu} radius of neutral fiber.

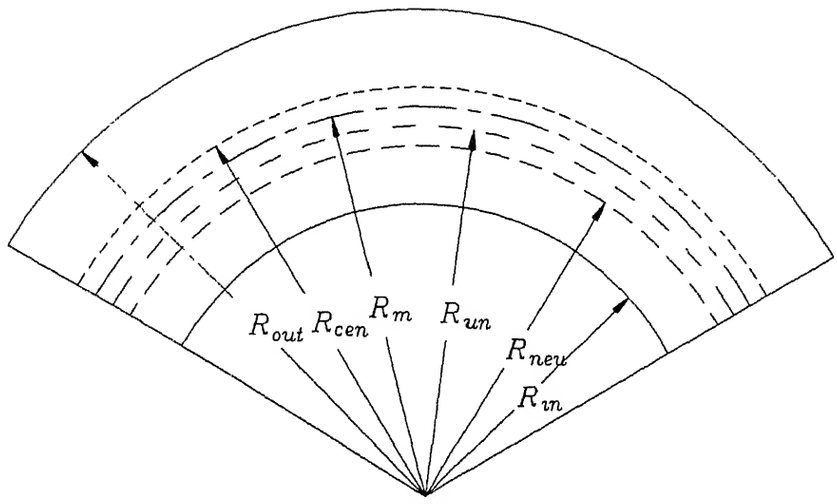


Figure 6 19 Location of important fibers in bending (Prasad and Somasundaram, 1993).

R_{out} outer radius of bend.
R_{un} radius of unstretched fiber.
R_{unnew} radius of new unstretched fiber.
t_d deformed sheet thickness.
t_{dnew} new deformed sheet thickness.
t_{in} initial sheet metal thickness.
σ_y yield stress.
θ included angle of bend.
θ_s start angle of bend.
θ_e end angle of bend.

Location of various important fibers in the bending process are shown in Fig. 6 19. Since strain hardening of the material is considered, these fibers coincide only at the beginning of the bending process. During bending process, fibers of the work material are subjected to transverse stress and strain conditions, and continuously change during the bending process Fibers between R_{out} and R_m are stretched during bending; fibers between R_m and R_{un} are initially compressed and then stretched more than they are compressed. Fiber at R_{un} represents the unstretched fiber and the stretch is equal to the compression. Fibers between R_{un} and R_{neu} are first compressed and subsequently stretched, but less than they were compressed initially. Fibers between R_{neu} and R_{in} are only compressed. The algorithm used to find the bend allowance (Prasad and Somasundaram, 1993) is given

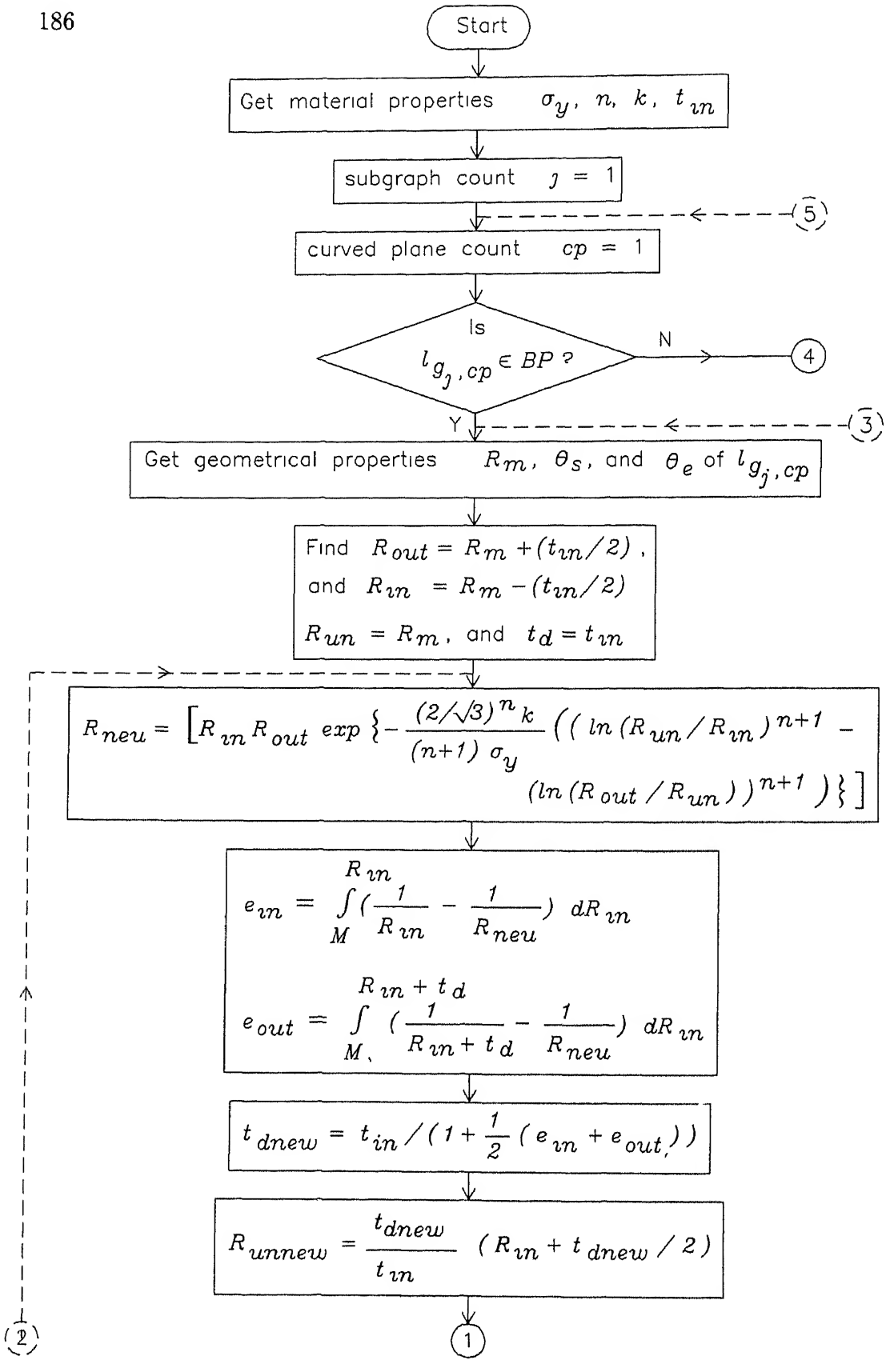


Figure 6.20: Flowchart for calculating bend allowance in the curved planes of a component.

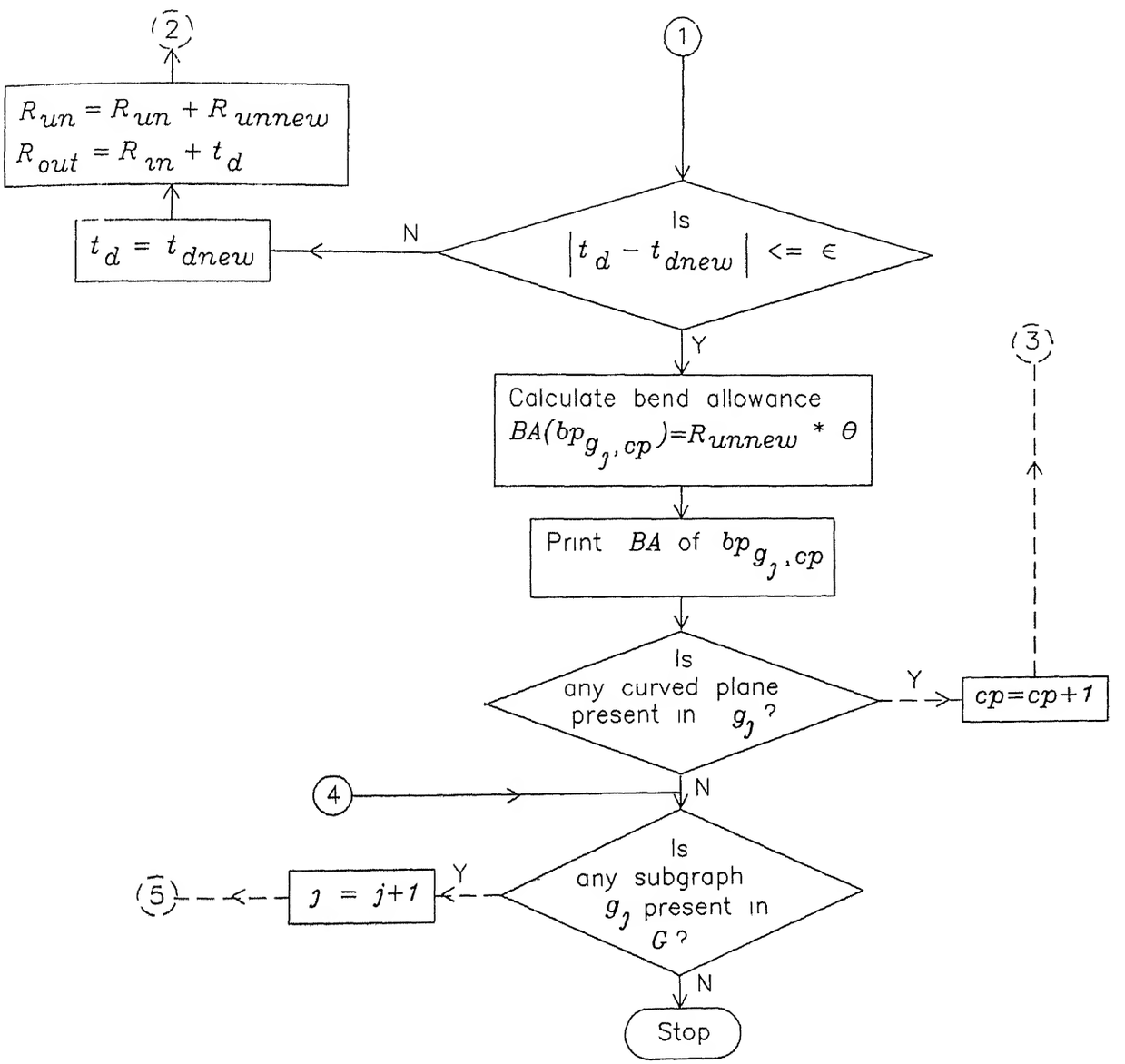


Figure 6.20: Flowchart for calculating bend allowance in the curved planes of a component (continued).

below ·

Algorithm and implementation

Step 1. Initially the surface strains are determined by assuming that the thickness remains constant, which is achieved by integrating the following equations

$$e_{in} = \int_{\infty}^{R_{in}} \left(\frac{1}{R_{in}} - \frac{1}{R_{neu}}\right) dR_{in} \tag{6 31}$$

$$e_{out} = \int_{\infty}^{R_{in}+t_d} \left(\frac{1}{R_{in}+t_d} - \frac{1}{R_{neu}}\right) dR_{in} \tag{6.32}$$

Step 2. The first approximation of the thickness of the sheet at the deformed zone is determined by using eqn. (6.33) with the value of strains in the previous step calculated

$$t_{dnew} = \frac{t_{in}}{(1 + \frac{1}{2}(e_{in} + e_{out}))} \tag{6 33}$$

Step 3. Again calculate the surface strains using the new value of the thickness of the sheet determined in the above step (Step 2) and eqns. (6.31) and (6.32).

Step 4. Goto step 3 until the difference of thickness (t_d) and t_{dnew} in the deformed zone in the successive approximation is below the prescribed limit.

The flowchart for calculating bend allowance along the unstretched fiber in a component is given in Fig. 6.20. Surface strains (e_{in} and e_{out}) are calculated at the evaluated value of R_{neu} by Gauss–quadrature numerical integration, assuming M (Fig.6.20) as a large value. The output of the flowchart for two curved planes present in the component shown in Fig 6.3(a), are given in Table 6.6 along with the inputs used for calculating the bend allowance.

6.8.2 Flat pattern development and implementation

The flattening process for a component begins by finding a flat face having maximum number of edges in a parent subgraph. The wireframe model is drawn such that it represents the mean axis of the component. The bend allowance (BA) is calculated as given in the previous section by taking thinning effect into consideration, and finding the radius of the unstretched fiber. Planes are flattened recursively with respect to the reference plane (Satyadev, 1995), by initially rotating the bend plane and the subsequent planes about

All dimensions in	mm	
Raw material type		A K Steel
Yield stress (kgf/mm ^{2})	σ_y	172 80
Strength coefficent	k	510 6
Strain hardening exp	n	0 22
Initial thickness	t_{in}	2 00
Plane		$bp_{g_1,1}$
Radius of mean fiber	R_m	13 83
Included angle	θ	55
Bend allowance	BA	13 15
Plane		$bp_{g_2,1}$
Radius of mean fiber	R_m	5 00
Included angle	θ	180
Bend allowance	BA	15 45

Table 6.6: **Bend allowance determined for a curved plane of the component graph shown in Fig 6.3(a).**

angle θ of the bend plane The points of the subsequent planes are then translated by an amount equal to the bend allowance (BA) multiplied by the direction cosines in the direction of development, such that the points of a planar adjacent to the bend plane under consideration lie in the reference plane. The process is repeated till all the points of the component are located in the reference plane, i e., 3–D component is mapped into a 2–D flat plane.

The process of flattening of a component (Fig 6 21(a)) is shown in (Fig. 6 21(b)) for its mean axis Let fp_1 be the reference plane, and $p_0(x_0, y_0, z_0)$ represent the first point through which, an arbitrary axis passes in space. Let θ be the included angle of the bend plane (bp_1) which has to be rotated for flattening the component, and the direction cosines are represented by c_x, c_y, c_z

Algorithm

Step 1. The bend plane (bp_1) under consideration, and the subsequent planes (fp_2, bp_2, fp_3) are rotated by θ about an arbitrary axis in the downward direction with respect to the reference plane (fp_1), since the bend plane is in the upward direction. Rotation about an arbitrary axis by an angle θ is accomplished using the following procedure

Step 1(a). Translate the system such that the point $p_0(x_0, y_0, z_0)$ is at the origin of the system.

Step 1(b). Perform appropriate rotations to make the axis of rotation coincident with the Z axis (Choice of Z axis is arbitrary).

Step 1(c). Rotate about Z axis by the included angel θ.

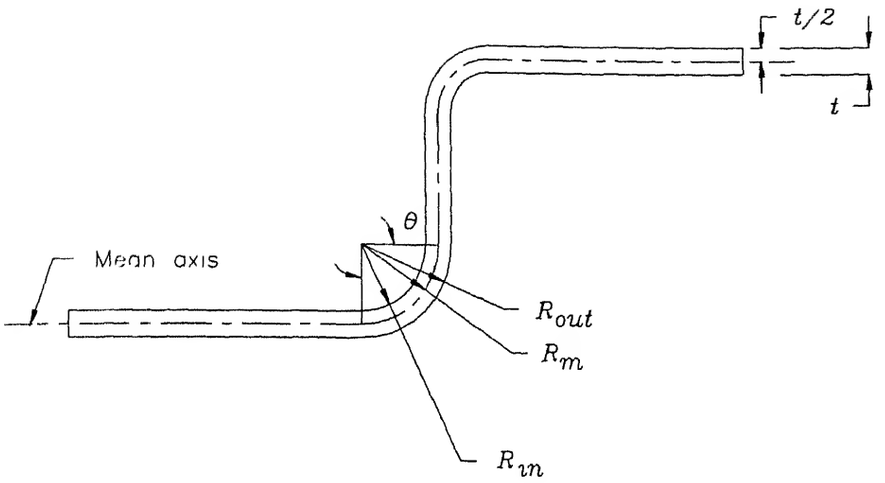
Step 1(d). Perform the inverse of combined rotation transformation.

Step 1(e). Perform the inverse of translation.

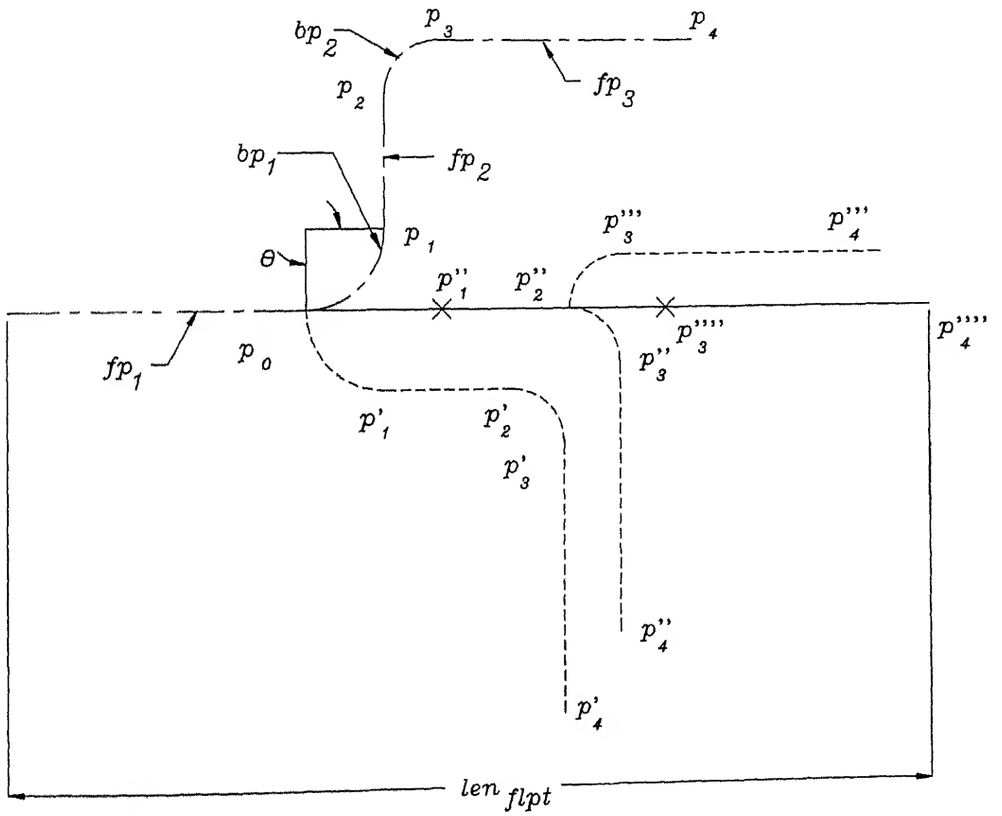
The complete transformation matrix ($[M]_{p_0}$) of points located in the subsequent planes (fp_2, bp_2, and fp_3 in Fig. 6.21(b)) about an arbitrary axis (Roger and Adams, 1990) in space is given by

$$[M]_{p_0} = [T][R_X][R_Y][R_{theta}][R_Y]^{-1}[R_X]^{-1}[T]^{-1} \tag{6.34}$$

In Fig. 6 21(b), the rotated points p_1, p_2, p_3, and p_4 are shown as p_1', p_2', p_3', and p_4'.



(a)



$$len_{flpt} = len_{fp_1} + len_{bp_1} + len_{fp_2} + len_{bp_2} + len_{fp_3}$$

(b)

Figure 6.21: (a) Mean axis of a bend, and (b) Flattening process along the mean axis of the bend.

Step 2. Points other than those lying on the reference plane and arbitrary axis are translated by the matrix of bend allowance M_{BA}, such that the points lying on the adjacent plane (fp_2) is in the reference plane and they are in the direction of development. The M_{BA} matrix is

$$M_{BA} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ x_0 - x_1 + BA * c_x & y_0 - y_1 + BA * c_y & z_0 - z_1 + BA * c_z & 1 \end{bmatrix} \quad (6\ 35)$$

where the directional cosine (c_x, c_y, c_z) is given by :

$$[c_x\ c_y\ c_z] = \frac{(x_1 - x_0)(y_1 - y_0)(z_1 - z_0)}{[(x_1 - x_0)^2 + (y_1 - y_0)^2 + (z_1 - z_0)^2]^{1/2}} \quad (6\ 36)$$

and the direction of flat pattern development is determined as given below :

Step 2(a). Determine a normal vector $\vec{NV}_{ref}$ at the point p_0 for the reference plane (fp_1).

Step 2(b). Find a development vector $\vec{V}_{dev}$, which is normal to the normal vector $\vec{NV}_{ref}$ and the arbitrary axis vector $\vec{V}_{arb}$. Development vector $\vec{V}_{dev}$ at p_0 is determined by the cross product of the above two vectors, i.e.,

$$\vec{V}_{dev} = \vec{V}_{arb} \times \vec{NV}_{ref}. \quad (6.37)$$

Step 2(c). Determine a vector $\vec{V}_{p_1}$ passing through the point p_1.

Step 2(d). The dot product of the vectors $\vec{V}_{dev}$ and $\vec{V}_{p_1}$ is determined. If the value of the dot product is positive, then the $\vec{V}_{dev}$ is in the direction of flat pattern development of the component, else the direction of development is in the opposite direction of the vector $\vec{V}_{dev}$.

In Fig. 6.21(b), points p_1'', p_2'', p_3'' and p_4'' represent the translated points using the bend allowance matrix (M_{BA}) for the points p_1', p_2', p_3' and p_4' respectively.

Step 3. Find another point for which the flattening process of planes is required.

If there is a bend plane, then select a point in the bend plane as the reference point through which the arbitrary axis passes, and continue the process of flat pattern development. If there is no bend plane, i.e., all the planes of the component are located in the reference plane, then the flat pattern development process of the component comes to an end.

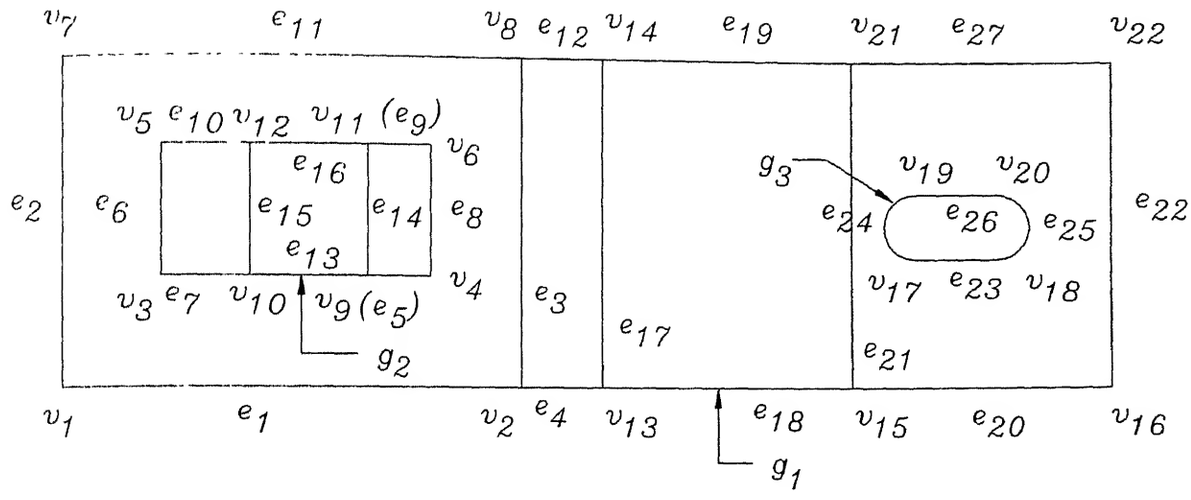


Figure 6.22. Flat pattern development of the 3–D component shown in Fig 6 3

Fig 6.21(b) gives the entire flat pattern development process by taking thinning effect into consideration The length of the flat pattern gives the length of the raw material required for the component.

Using the flat pattern development methodology, the flat pattern of the component shown in Fig. 6.3(a) is given in Fig. 6 22 Vertices and edges associated with the flat pattern with respect to Fig. 6.3(b) are also given in Fig. 6.22. It should be noted that the edge e_9 is from vertex v_5 to v_6 in the child subgraph g_2 This flat pattern of 3–D component is further used to recognize shearing features present in a component and/or component layout.

6.9 Recognition of shearing features

As stated earlier shearing operations are those, in which the work material is stressed beyond its elastic limit, cross sectional area is reduced, and fractured through the cross sectional area, when the ultimate strength is exceeded. The form features produced on the material are of 2–D. Recognition of shearing features in a nested layout has been discussed in the Chapter 5. As stated earlier, shearing features are classified into three categories viz., inside features, boundary features, and outside features Inside features of a 3–D component can be recognized from the individual component, if the created component from wireframe model is in accordance with Euler's formula Boundary and outside features are recognized from the optimized component layout created from the flat pattern of a 3–D component.

In the following section, procedural details for the creation of 2–D component using the flat pattern of 3–D component are given, such that the vertices and edges present in the developed 2–D component layout are in accordance with Euler's formula

6.9.1 Creation of the 2–D component from the flat pattern of a 3–D component

The flat pattern developed will have common edges, and may have coincident edges. Coincident edges are those edges, wherein an edge coincides with another edge or forms a part of an another edge. In Fig. 6.3, e_7 and e_{13} falls on the edge e_5 when a louvering feature is developed (Fig 6.22). To recognize shearing features in the flat pattern developed, common edges and coincident edges have to be removed from the flat pattern of the component The methodology followed for the creation of 2–D component boundary and internal features is discussed below .

Creation of boundary feature of a component

The parent subgraph of a 3–D component represents the boundary of the component. Therefore, flat pattern representing the parent subgraph is processed for creating the boundary of a component in 2–D. The boundary of the component in 2–D is obtained by deleting the common edges present in it. Identification of common edges is given in Section 6.3.1. Once the common edges are deleted, the adjacent edges incident on the start and end vertices of the common edges and which are adjacent to each other meet at a vertex. These edges violate the definition of a vertex when the edges are collinear. Therefore, these edges are replaced by a single new edge. Thus, a 2–D component boundary is created

Fig. 6.22 gives the flat pattern of the component shown in Fig. 6.3(a), along with the vertices and edges labelled. In this flat pattern developed, g_1 represents the flat pattern of the parent subgraph of the component shown in Fig. 6.3(a). In Fig. 6.23, the 2–D boundary of the component created after deleting the common edges (e_3, e_{17} and e_{21}), and replacing the adjacent edges (e_1, e_4, e_{18} and e_{20}) of deleted common edges by a new edge ne_1 is shown. Old edges are replaced by ne_1, since they meet at vertices v_2, v_{13}, and v_{15} which violate the definition of a vertex as the edges are collinear. For the same reasons, adjacent edges (e_{11}, e_{12}, e_{19} and e_{27}) of the deleted common edges are replaced by a new single edge ne_2.

It should be noted that if cross bend feature is present in a parent or child subgraphs of a component, then after removing the common edges from the subgraph, the cross bend feature represents an internal feature in the component.

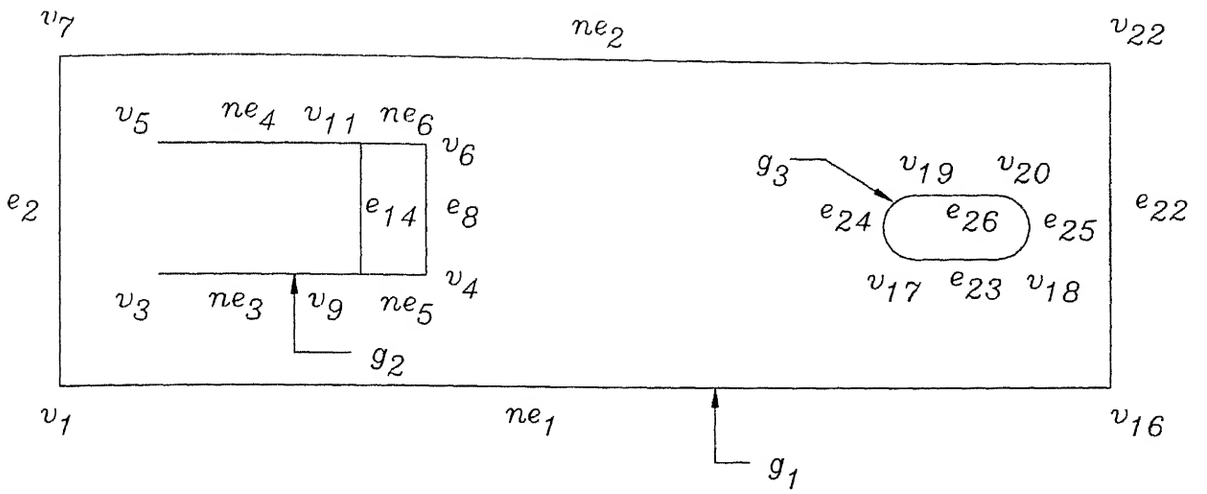


Figure 6 23. Creation of boundary and inside features from the flat pattern development given in Fig 6 22

Creation of inside features of a component

The child subgraphs of a component represent the inside features since they are located inside parent subgraphs. Also, as explained in the the previous section, cross-bend features transform into inside features Child subgraphs having only one loop represent an inside feature with number of edges and vertices same as that of subgraph. On the other hand, if it has more than one loop, then it may represent single inside feature or more As in the case of boundary feature development of a component from the parent subgraph, in child subgraphs development also, the edges adjacent to the deleted common edges meet at a vertex such that they violate the definition of a vertex. Therefore, these edges are replaced by a single new edge, if they are collinear.

In Fig 6.22 consider the flat pattern development of g_2 child subgraph. In g_2 child subgraph (Fig. 6.7(a)), there is more than one loop, hence the flat pattern has been developed. In this flat pattern also there are common edges. These common edges (e_6 and e_{15}) are deleted from the flat pattern. Collinear edges e_7 and e_{13} meet at v_{10}, and hence it violates the definition of a vertex. Therefore, v_{10} is deleted and in turn edges e_7 and e_{13} are replaced by a new edge ne_3. Similarly edges e_{10} and e_{16} are adjacent to the deleted common edge e_{15}, and is collinear. Hence, they are replaced by another new edge ne_4.

In Fig 6.22, edge e_5 is incident on v_3, and new edge ne_3 is a part of it Therefore, e_5 and ne_3 are coincident edges, since the edge e_5 is passing through the vertex v_9 and is incident on the vertex v_4 and edge ne_3 has start and vertices as v_3 and v_9. Therefore, edge e_5 is deleted from the graph, and is replaced by a new edge (ne_5), such that v_4 and v_9

are its start and end vertices Similarly, e_9 is deleted and a new edge ne_6 is replaced such that, v_6 and v_{11} are its start and end vertices. Thus, Fig. 6.23 shows the creation of inside feature from the flat pattern (Fig 6 22) of child subgraph g_2. The vertices and edges of g_2 in Fig 6 23 adhere to Euler's formula In case of g_3 child subgraph, there is only one loop, and hence common edges are absent. Therefore, the created inside feature g_3 (Fig 6.23) is similar to that of the flat pattern of g_3 shown in Fig. 6 22

6.9.2 Recognition of inside features of a 3–D component from its flat pattern

If a child subgraph has only one loop, then the loop represents a piercing feature (recognition of piercing feature is discussed in Chapter 5). On the other hand, if it has more than one loop, then the subgraph may have both piercing and lancing features or only a lancing feature If the child subgraph is represented by more than three edges while creating the inside feature for a component, then it represents that the child subgraph has both lancing and piercing feature. On the other hand, if there are only three edges, then the child subgraph has only a lancing feature. Piercing and lancing features are recognized as explained in the section on recognition of inside shearing features in Chapter 5.

In Fig 6 23, the child subgraph (g_2) has five edges, and hence this has both lancing and piercing features. Loops in the created subgraph are identified after deleting edges (ne_3 and ne_4) which are incident on the vertices (v_3 and v_5) of degree one. Afterwards, in accordance with the recognition of inside shearing features of a component, it has been recognized that vertices (v_9, v_4, v_{11} and v_6) and edges (ne_5, e_{14}, e_8 and ne_6) form a loop of vertices and edges of piercing feature, and vertices (v_3, v_9, v_{11} and v_5) and edges (ne_3, e_{14} and ne_4) form a loop of vertices and edges of lancing feature.

6.9.3 Recognition of boundary and outside features of a 3–D component from its flat pattern

Component nested layout is used for recognizing boundary and outside features.Component nested layout is created with the help of the 2–D component created from the flat pattern developed for a 3–D component (i.e., output of Section 6 8.1). With the output of vertices and edges of Section 6.8.1, a wireframe model representing the 2–D component is created in the CAD database (in the present work, DXF input file for AuotoCAD Release-10 is created, and is used to create the 2–D component). Using this wireframe model, nested component layout is manually created by the designer. As explained in Chapter 5, the output file of CAD database is used to recognize boundary and outside shearing features.

Chapter 7

Prototype System and Implementation

7.1 Introduction

In order to provide the demonstration of the concepts used in the present work, a prototype system is developed and is described below ·

The system is coded in the C language and it is implemented on both PC/AT and HP 9000/850. Various major functions of the prototype system are shown in Fig. 7.1. The over all control of the system is given in the flowchart shown in Fig. 7.2. Input used to the system is Drawing interchange (DXF) output file of AutoCAD (Release-10). DXF is an ASCII file and the various sections present in the DXF are given in Table 7.1. Representation for an entity (line) in the DXF is given in Table 7.2.

The preprocessor of the system processes the component/part layout data created in wireframe geometric modeler, and the data relevant to the entities are extracted. If the data consists of working or entity coordinate system (ECS), then it is converted into world coordinate system (WCS). These data are used for identifying the nature of an input (i e 2–D or 3–D component) system to recognize pressworking features and for flat pattern development.

The CAD database of the AutoCAD gives the extrusion direction of the entity arc and circle, when they are drawn in other than XY plane. This helps in reducing the space for representing the entities in CAD database, and with the help of 3–D vector describing the Z axis of ECS, the information about the entity in 3–D space can be obtained. This is due to the fact that, along the Z axis direction, there can be only one ECS, since the properties are associated with it.

1. The origin coincides with the WCS origin.

2. The orientation of X and Y axes with the XY plane are calculated in an arbitrary, but in a consistent manner.

```
  0          (Begin HEADER section)
SECTION
  2
HEADER
     [Header variable items go here]
  0
ENDSEC   (End HEADER section)
  0          (Begin TABLE section)
SECTION
  2
TABLES
  0
TABLE
  2
VPORT
  70
     (Viewport table maximum item count)
     [Viewport table items come here]
  0
ENDTAB
  0
TABLE
  2
     LTYPE, LAYER, STYLE, VIEW, UCS, DWGMGR
  70
     (Table maximum item count)
     [Table items come here]
  0
ENDTAB
  0
ENDSEC   (End TABLE section)
  0          (Begin BLOCKS section)
SECTION
  2
BLOCKS
  0
BLOCK
     [Block definition entities come here]
  0
ENDSEC   (End BLOCKS section)
  0          (Begin ENTITIES section)
SECTION
  2
ENTITIES
     [Drawing entities come here]
  0
ENDSEC   (End of ENTITIES section)
  0
EOF
```

Table 7.1: DXF output file format representing the various sections present in the file.

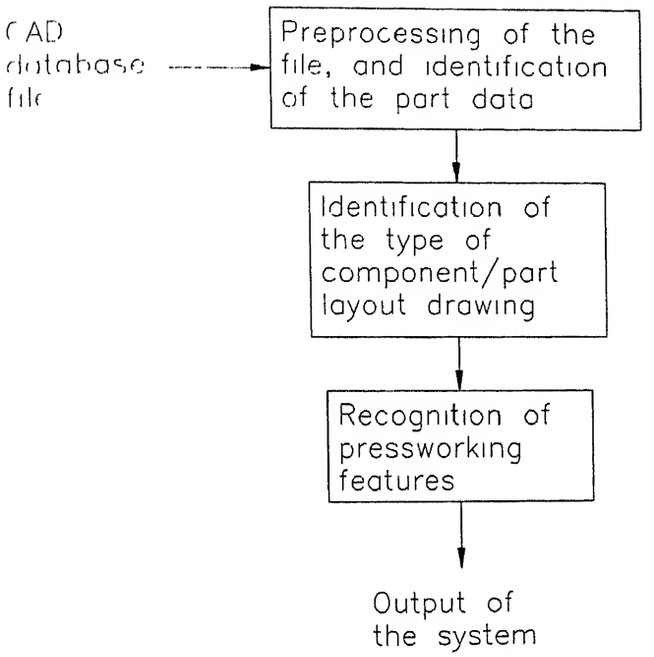


Figure 7.1: Various Functions of the prototype system.

0	Begin of entity
LINE	Entity name
8	Layer
0	Layer number
10	Start x
0.0	x coordinate value
20	Start y
0.0	y coordinate value
30	Start z
20.0	z coordinate value
11	End x
10.0	x coordinate value
21	End y
20.0	y coordinate value
31	End z
50.0	z coordinate value

Table 7.2: Explanation of an entity format given in DXF output file

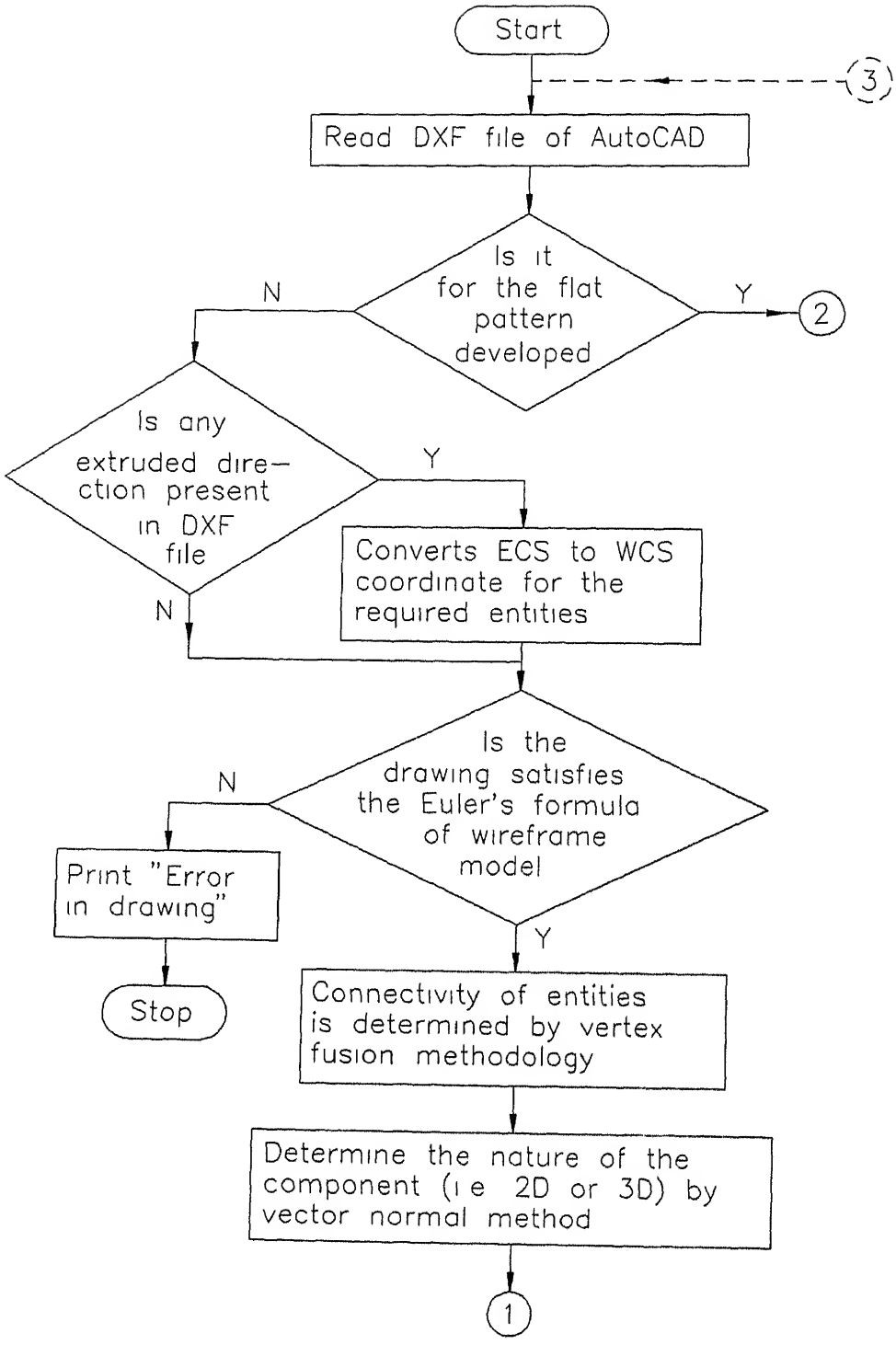


Figure 7.2: Flowchart representing the flow of control for the prototype system.

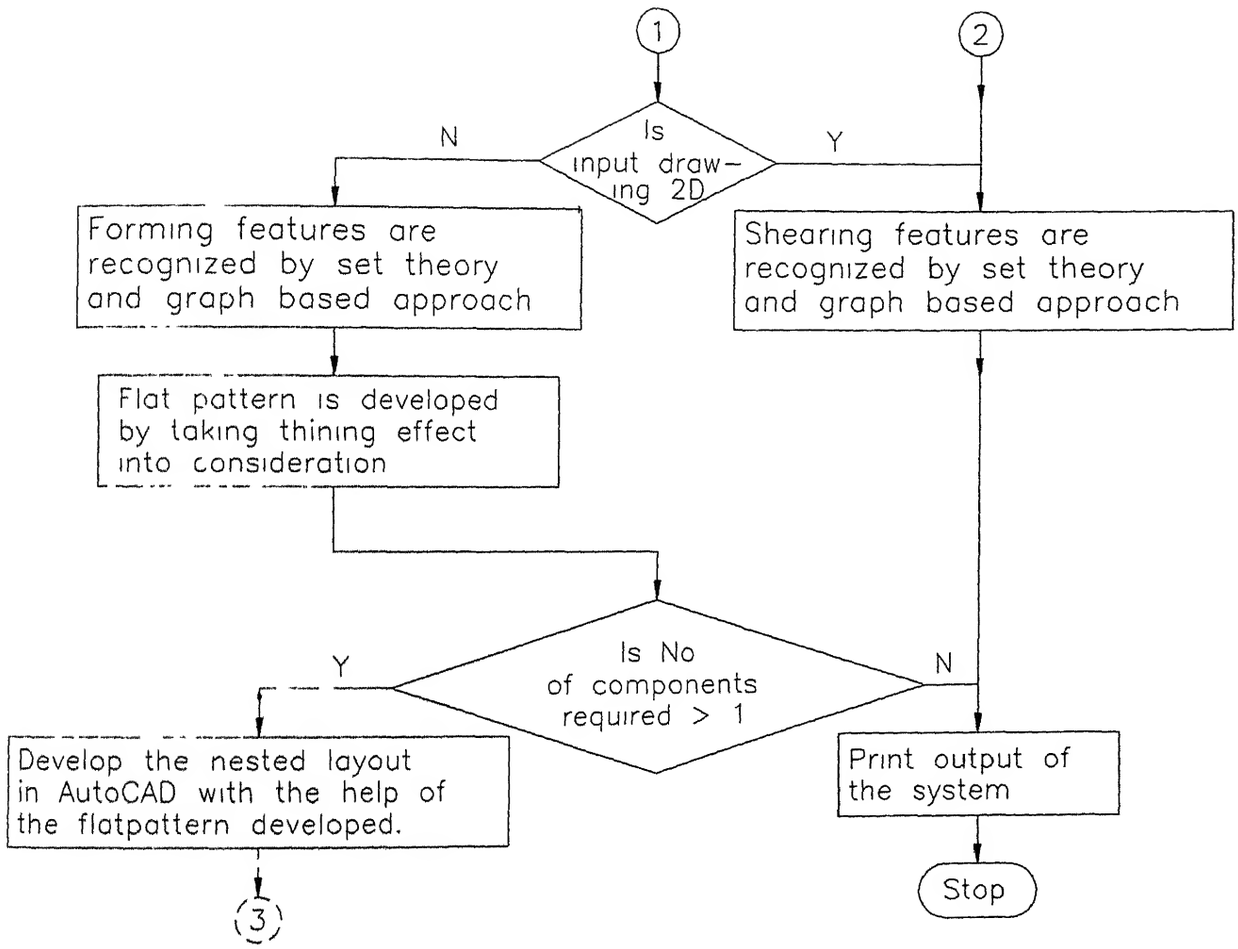


Figure 7.2: Flowchart representing the flow of control for the prototype system (continued).

The algorithm called *arbitrary axis algorithm* (AutoCAD, Release-10, 1989) is used to convert the ECS of an entity to WCS. After the identification of the nature of the input to the system (i e. 2–D or 3–D component) with the help of graph theory, the control is transferred to the appropriate recognition process.

The various functions for the recognition of shearing features, and the data structure type used for recognizing them are given in Figs. 7 3 and 7.4, respectively. Similarly, various functions for recognizing the forming features, and the data structure type used for recognizing the same are given in Figs 7 5, and 7.6 respectively.

7.2 Implementation

In order to have a better understanding, and to demonstrate the overall effectiveness of the concepts developed in this research, real life components are considered and results/output (given at the end of this chapter) for a few tested components/layout are given below :

Components of 2–D in nature

Example–1 : A 2–D Bezel part layout for *similar components* (Fig. 7.7(a)) used for manufacturing of a watch cases at HMT Watch Factory, (India), is given (number of components at present considered in the layout are only five, but in the factory depending upon the size usually ten to twelve components are nested in a strip type of raw material). The plan view of the layout is represented in the wireframe model. The list of vertices and entities after lexicographical ordering are given in the Table 7 3 and Table 7.4 respectively. Also in the Example–1 list, the output of the system is given. To understand the output, a brief representation of vertices, edges, MER for components, features and a few shearing feature types are given in Fig 7 7(b).

Example–2 : Another 2–D part layout for *dissimilar components* used in the manufacturing of Vertical Machining Center–345 (VMC–345) at The Mysore Kirloskar Limited, Harihar (India) is shown in the Fig. 7.8(a). The list of vertices and edges after lexicographical ordering is given in the Table 7 5 and Table 7 6 respectively The output of the system is given in Example–2, and for easy understanding, vertices and features are represented in Fig.7 8(b).

Components of 3–D in nature

Example–3 : A simple 3–D component's mean plane (Fig.7.9(a)) to demonstrate the extraction of cross bend feature, flat pattern development, identification of bending, curling and hemming features is given. Various plane numbers of the component is given in Fig 7.9(b) for easy reference. The flat pattern developed for this simple 3–D component is given in Fig. 7.10. The list of vertices and the list of edges present in the CAD database

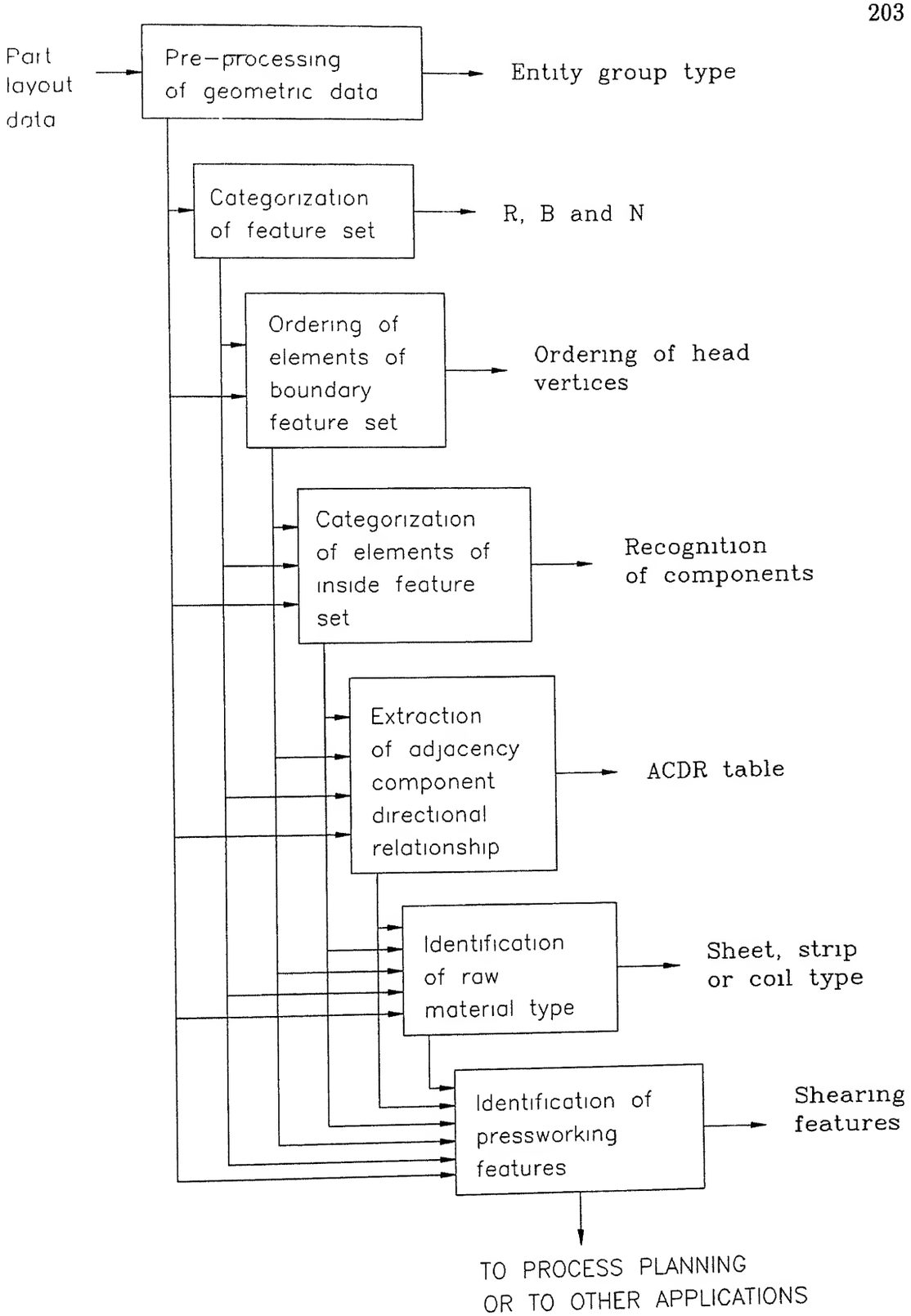


Figure 7.3· Various functions of shearing feature recognition system

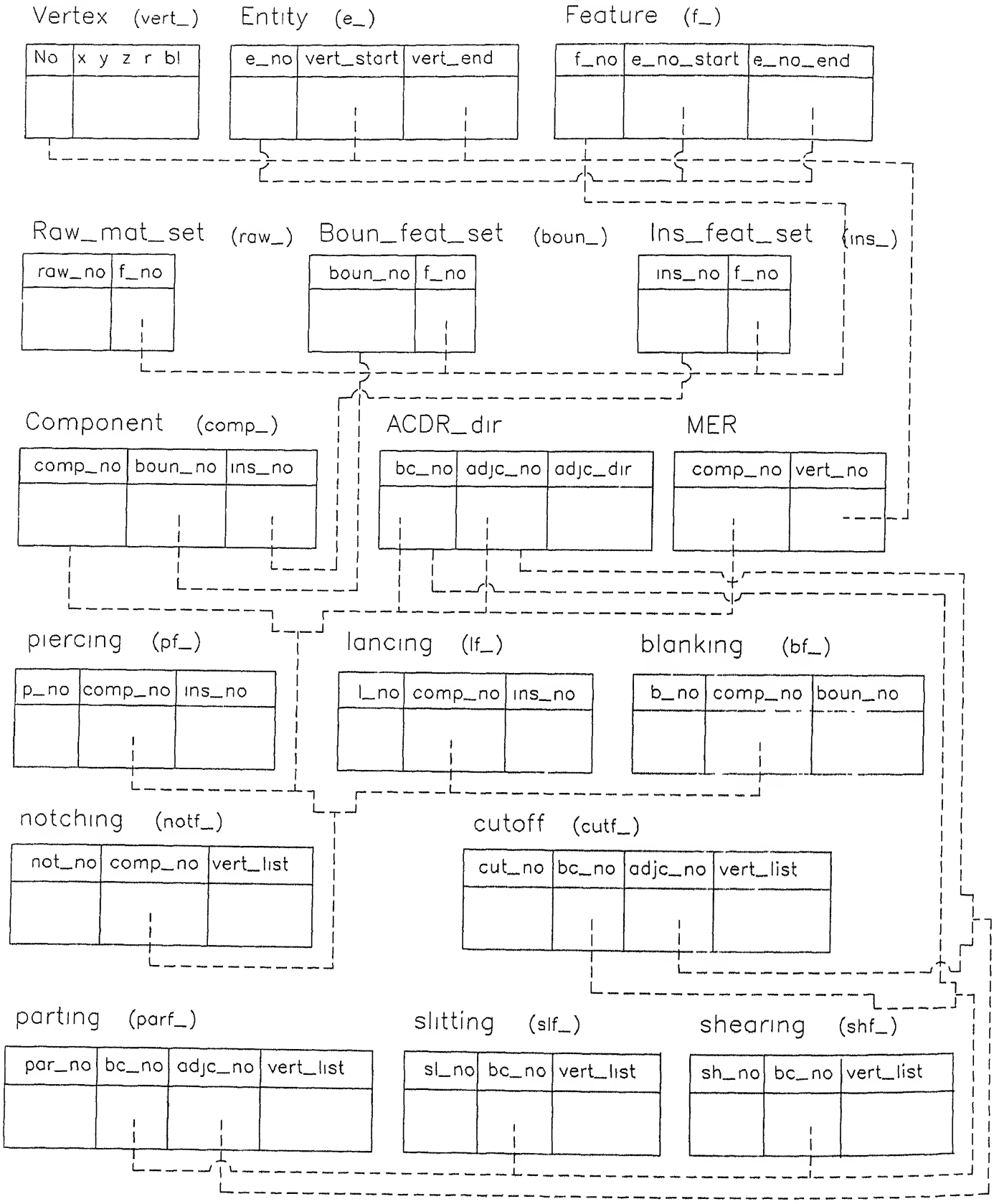


Figure 7.4: Operation based data structure for recognizing shearing features.

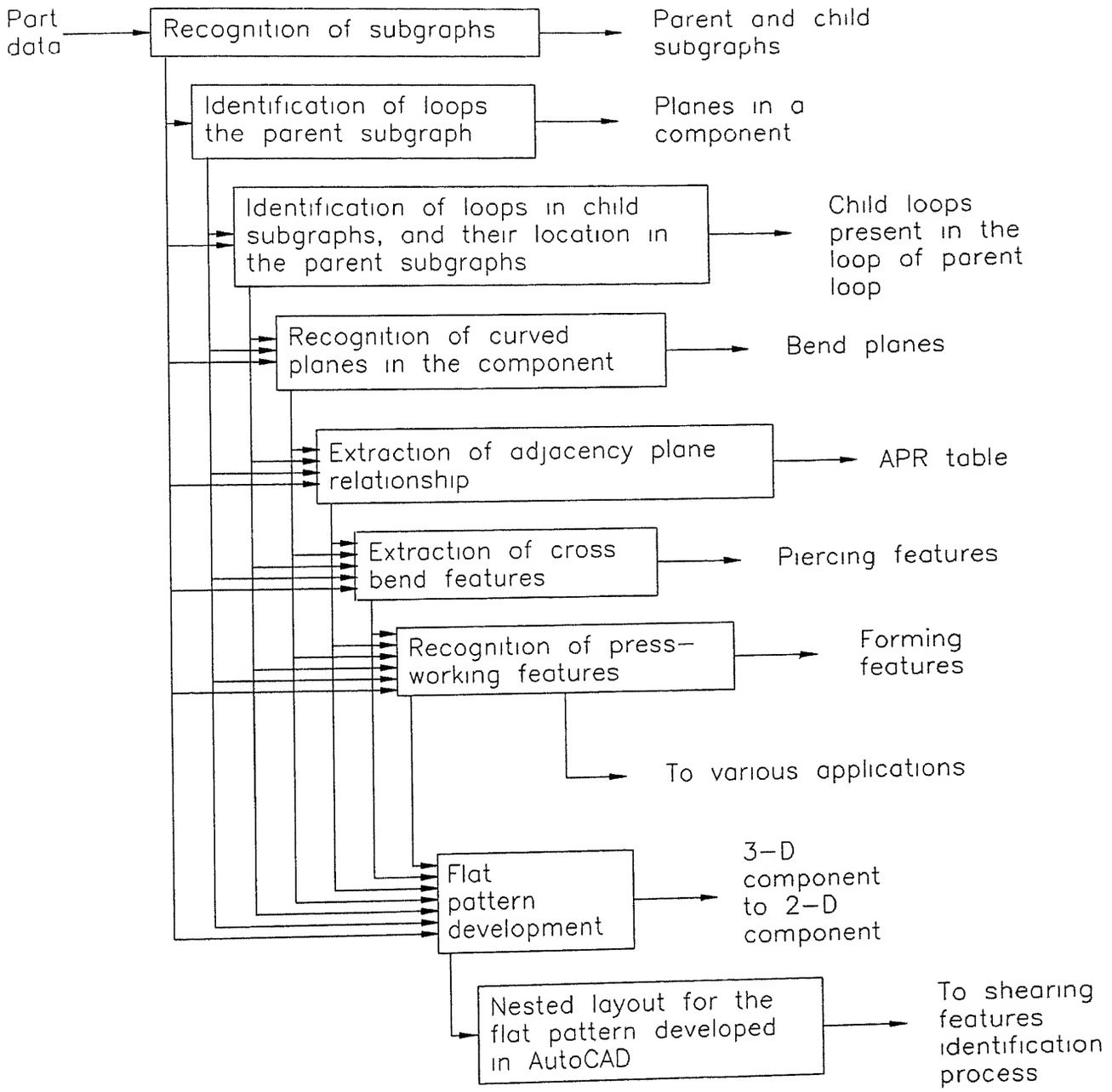


Figure 7.5: Various functions of forming feature recognition system

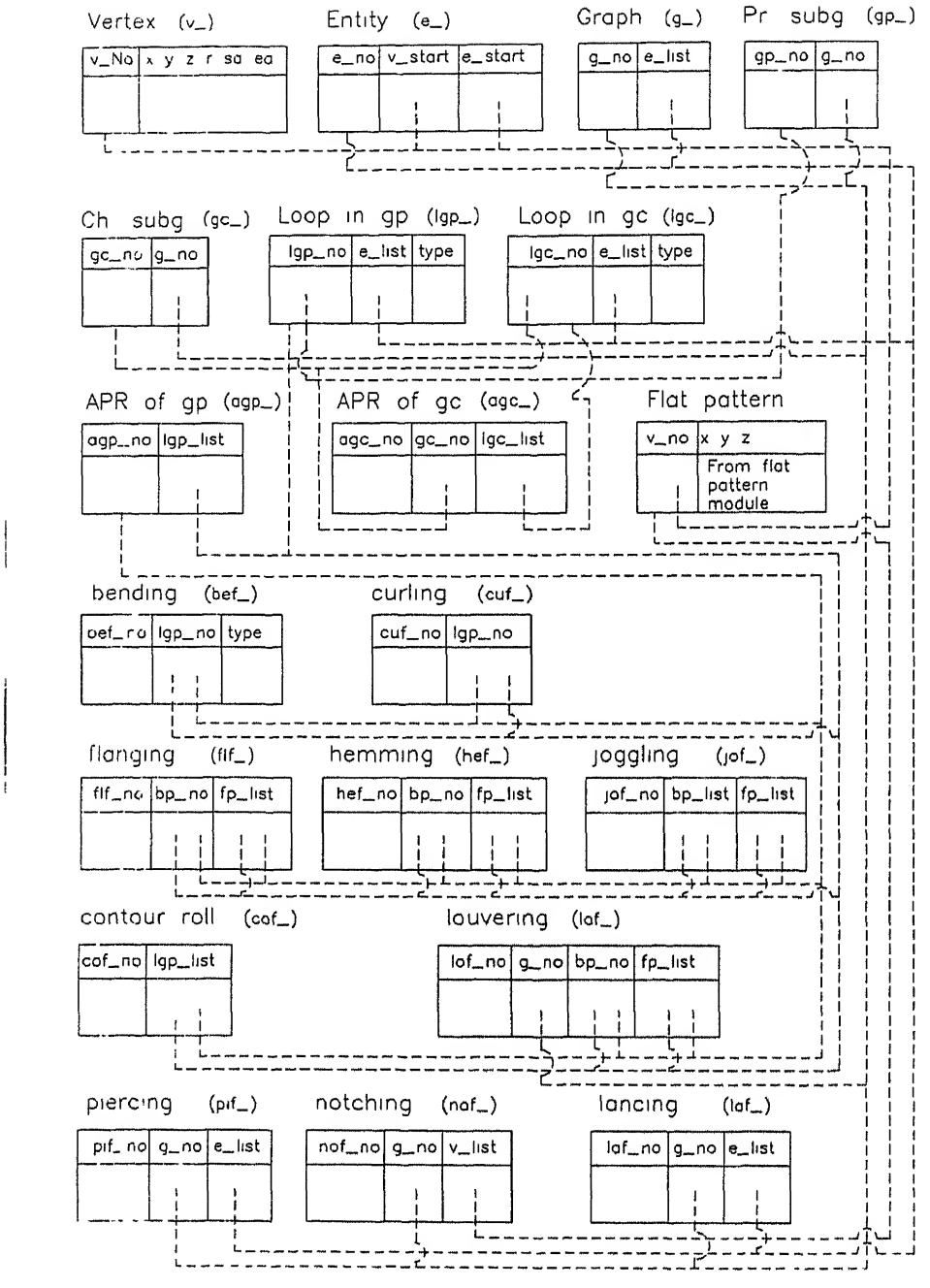


Figure 7.6: Operation based data structure for recognizing forming features.

after removing duplicate vertices and lexicographical ordering are given in Table 7 7 and Table 7 8 respectively The output of the system is given in Example–3

Example–4 : A part used in the box of sun microphone of a modified complex Sun Microsystems Computer Corporation Inc for keeping the microphone is shown in Fig. 7 11(a), and various planes present in it given in Fig 7 11(b) for easy reference The flat pattern developed for this complex object is given in Fig 7 12 The lexicographically ordered list of vertices and the list of edges are given in Table 7 9, and Table 7 10 respectively. The output of the system for this component is given in Example–4.

Example–1

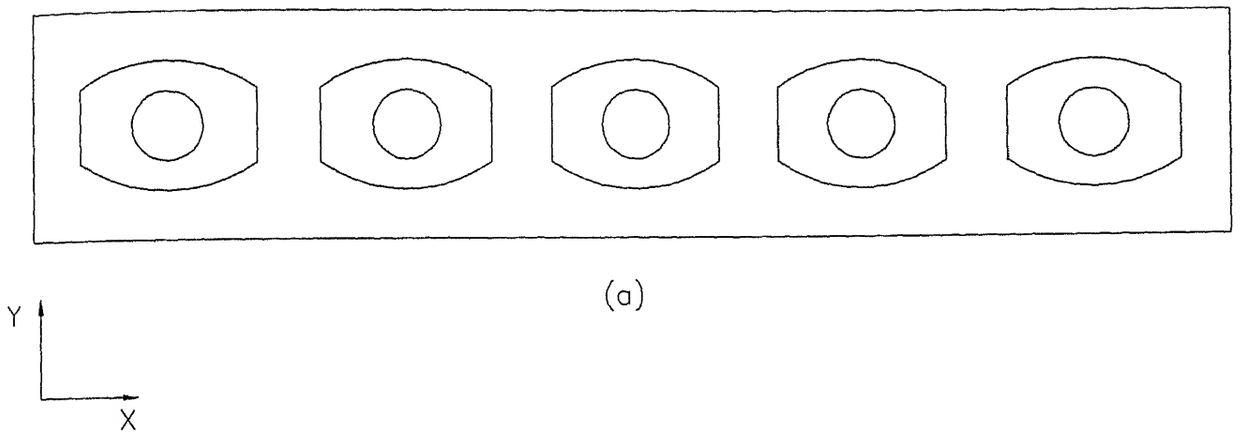


Figure 7.7: Bezel layout of a watch manufactured at HMT (India) (a) Drawing of the bezel nested layout given as input to the system, and (b) Bezel layout giving a part of information about the output for easy reference

Vert No	x	y	rad /bul	start ang	end ang
1	30 0000	205 0000	–	–	–
2	275 0000	205 0000	–	–	–
3	40 0004	220 4997	–	–	–
4	75 0002	220 5001	30 0000	234 3150	305 6850
5	87 5004	220 4997	–	–	–
6	122 5002	220 5001	30 0000	234 3150	305 6850
7	135 0004	220 4997	–	–	–
8	170 0002	220 5001	30 0000	234 3150	305 6850
9	182 5004	220 4997	–	–	–
10	217 5002	220 5001	30 0000	234 3150	305 6850
11	230 0004	220 4997	–	–	–
12	265 0002	220 5001	30 0000	234.3150	305.6850
13	57 5000	228.0000	7 0000	–	–
14	105 0000	228 0000	7 0000	–	–
15	152 5000	228 0000	7 0000	–	–
16	200 0000	228 0000	7 0000	–	–
17	247 5000	228 0000	7 0000	–	–
18	40 0000	235 5000	30 0000	54 3150	125 6850
19	75 0000	235 5000	–	–	–
20	87 5000	235 5000	30 0000	54 3150	125 6850
21	122 5000	235 5000	–	–	–
22	135 0000	235 5000	30 0000	54.3150	125 6850
23	170 0000	235 5000	–	–	–
24	182 5000	235 5000	30 0000	54 3150	125.6850
25	217.5000	235 5000	–	–	–
26	230 0000	235.5000	30.0000	54 3150	125.6850
27	265.0000	235 5000	–	–	–
28	30 0000	251 0000	–	–	–
29	275 0000	251 0000	–	–	–

Table 7.3: List of vertices after lexicographical ordering.

Edge No	Entity type	Start vert No	End vert No
1	4	1	2
2	4	1	28
3	4	2	29
4	3	3	4
5	4	3	18
6	4	4	19
7	3	5	6
8	4	5	20
9	4	6	21
10	3	7	8
11	4	7	22
12	4	8	23
13	3	9	10
14	4	9	24
15	4	10	25
16	3	11	12
17	4	11	26
18	4	12	27
19	1	13	–
20	1	14	–
21	1	15	–
22	1	16	–
23	1	17	–
24	3	18	19
25	3	20	21
26	3	22	23
27	3	24	25
28	3	26	27
29	4	28	29

Table 7.4: List of edges after lexicographical ordering

part	:	Bezel layout of a watch manufactured at HMT
material	:	Aluminum
units	:	mm
thickness	:	6 00
Component/ layout type	:	2-D
No. of features	:	11

Vertices present in various features :

Feature No	Vertex No and its degree			
1	1 (2)	2 (2)	28 (2)	29 (2)
2	3 (2)	4 (2)	18 (2)	19 (2)
3	5 (2)	6 (2)	20 (2)	21 (2)
4	7 (2)	8 (2)	22 (2)	23 (2)
5	9 (2)	10 (2)	24 (2)	25 (2)
6	11 (2)	12 (2)	26 (2)	27 (2)
7	13 (0)			
8	14 (0)			
9	15 (0)			
10	16 (0)			
11	17 (0)			

Raw material feature set (R)	:	$\{1\}$
Boundary feature set (B)	:	$\{2,3,4,5,6\}$
Inside feature set (N)	:	$\{7,8,9,10,11\}$
Component set (C)	:	$\{1,2,3,4,5\}$
Inside features of a comp.	:	$c_1 = 1, c_1 = \{7\}$, $c_2 = 2; c_2 = \{8\}$, $c_3 = 3, c_3 = \{9\}$, $c_4 = 4, c_4 = \{10\}$, $c_5 = 5; c_5 = \{11\}$,

Pseudo vertices :

Vert No	x	y
4'	57 500	214 867
18'	57 500	241 133
6'	105 000	214 867
20'	105 000	241 133
8'	152 500	214 867
22'	152 500	241 133
10'	200 000	214.867
24'	200 000	241 133
12'	247.500	214 867
26'	247 500	241 133

MER of components :

Comp No	Lower left vertex		Upper right vertex	
	x	y	x	y
1	40 000	214 867	75 000	241 133
2	87 500	214 867	122 500	241 133
3	135 000	214 867	170 000	241 133
4	182 500	214 867	217 500	241 133
5	230 000	214 867	265 000	241 133

ACDR Table :

c_j / c_j	1	2	3	4	5
1	-	1^{1}			
2	1^{3}	-	1^{1}		
3		1^{3}	-	1^{1}	
4			1^{3}	-	1^{1}
5				1^{3}	-

Raw material :

type	strip	
direction	X axis	
feature No.	1	
size	245 000 46 000	
position	30 000 205 000	275 000 251 000

Component No. : 1

Inside Features

piercing : 1
feature No. : 7
position : 57.500 228 000 7 000

Boundary Features

blanking :
shape : outline boundary of the component
feature no 2
position : 40 000 220 500
40 000 235 000 30 000 54 315 125 685
75 000 220.500
40 000 220 500 30.000 234 315 305 685

notching : 1
position 57.500 214 867 30 000 234.315 270 000
40.000 220 500
40 000 214 867

notching : 2
position 40 000 235 500 30 000 90 000 125 685
57 500 241 133
40 000 241 133

notching : 3
position 57 500 241 133 30 000 54 315 90 000
75 000 235 500
75 000 241 133

notching : 4
position 75 000 220 500 30 000 270 000 305 685
57 500 214 867
75 000 214 867

blanking with notching :
position 40 000 220 500
40 000 235 500 30 000 54 315 125 685
75 000 235 500
75 000 220 500 30 000 234 315 305 685

Note : Piercing, blanking, notching, and blanking with notching
are not given for other components, since the output is of similar type.

parting : 1
position 75 000 205 000
75 000 251 000
87 500 251 000
87 500 205 000

parting : 2
position 122 500 205 000
122 500 251.000
135.000 251.000
135 000 205.000

parting : 3
position 170 000 205 000
170 000 251 000
182 500 251 000
182 500 205 000

parting : 4
position 217 500 205 000
217 500 251 000
230.000 251 000
230 000 205 000

Outside Features

shearing : 1
direction . Y axis
range 10 000
position 30 000 205 000 40 000 205 000
30 000 251 000 40 000 251 000

shearing : 2
direction Y axis
range 12 500
position 75 000 205 000 87 500 205 000
75 000 251 000 87 500 251 000

shearing : 3
direction Y axis
range 12.500
position 122 500 205 000 135 000 205 000
122 500 251 000 135 000 251 000

shearing : 4
direction Y axis
range 12 500
position 170 000 205 000 182 500 205 000
170.000 251.000 182 500 251 000

shearing : 5
direction . Y axis
range : 12.500
position 217 500 205.000 230 000 205.000
217 500 251.000 230 000 251 000

shearing : 6
direction Y axis
range : 10 000
position : 265.000 205 000 275.000 205 000
265 000 251 000 275 000 251 000

shearing : 7
direction : X axis
range 9.867
position 30.000 205.000 30 000 214 867
275.000 205.000 275 000 214 867

shearing : 8
direction . X axis
range . 9 867
position : 30.000 241 133 30 000 251 000
275 000 241.133 275 000 251 000

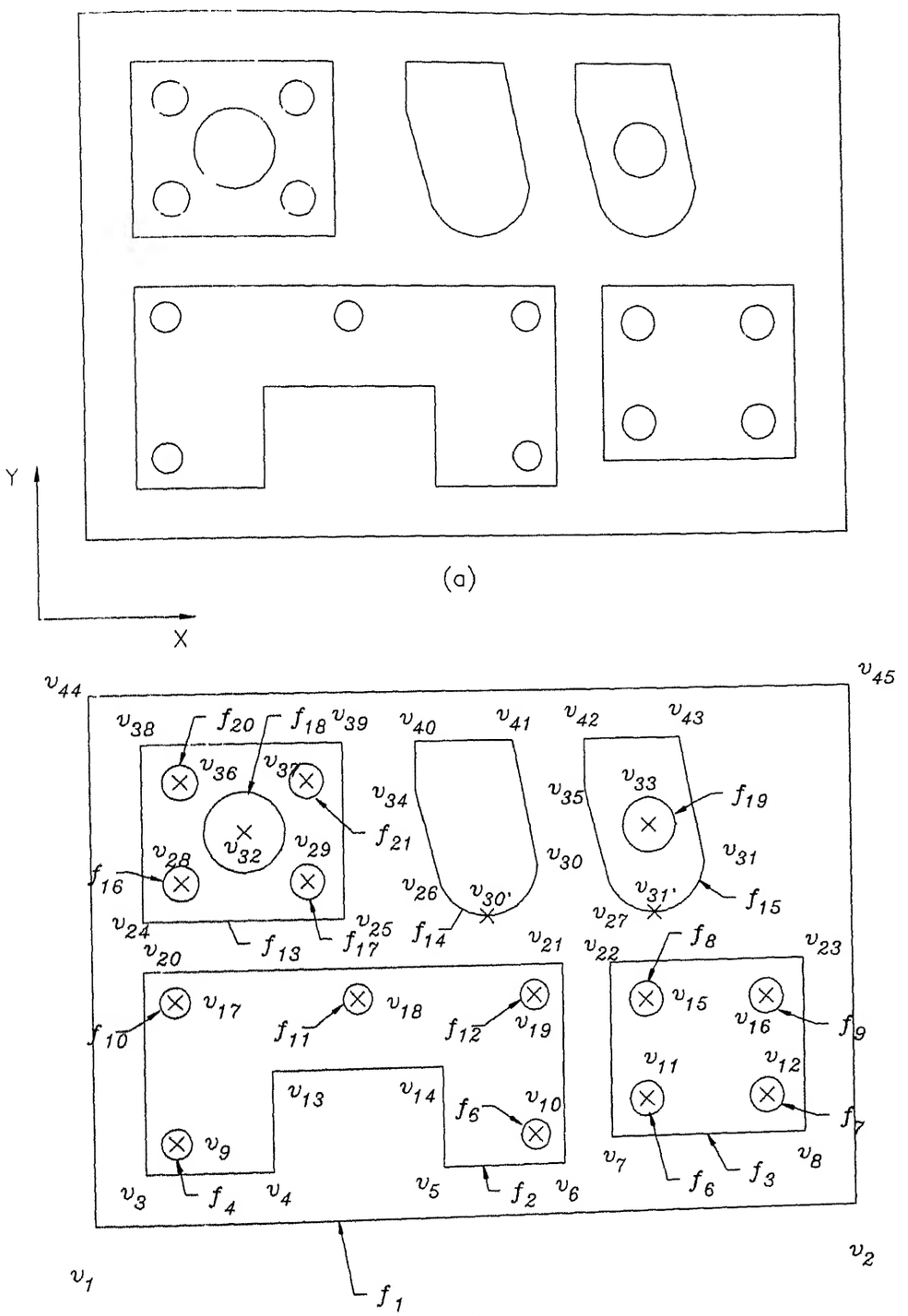


Figure 7.8: Dissimilar components used in the manufacturing of VMC–345 . (a) Drawing of the layout given as input to the system, and (b) Dissimilar components layout giving a part of information about the output for easy reference.

Vert No	x	y	rad /bul	start ang	end ang
1	30 0000	195 0000	–	–	–
2	185 0000	195 0000	–	–	–
3	40 0000	205 0000	–	–	–
4	65 0000	205 0000	–	–	–
5	100 0000	205 0000	–	–	–
6	125 0000	205 0000	–	–	–
7	135 0000	210 0000	–	–	–
8	175 0000	210 0000	–	–	–
9	46 0000	211 0000	3 0000	–	–
10	119 0000	211 0000	3 0000	–	–
11	142 5000	217 5000	3.5000	–	–
12	167 5000	217 5000	3 5000	–	–
13	65 0000	225 0000	–	–	–
14	100 0000	225 0000	–	–	–
15	142 5000	237 5000	3 5000	–	–
16	167 5000	237 5000	3 5000	–	–
17	46 0000	239 0000	3.0000	–	–
18	82 5000	239 0000	3 0000	–	–
19	119 0000	239 0000	3.0000	–	–
20	40 0000	245 0000	–	–	–
21	125 0000	245 0000	–	–	–
22	135 0000	245.0000	–	–	–
23	175 0000	245.0000	–	–	–
24	40.0000	255 0000	–	–	–
25	80 0000	255 0000	–	–	–
26	100.0000	260 9999	–	–	–
27	135 0000	260.9999	–	–	–
28	47 5000	262.5000	3.5000	–	–
29	72.5000	262 5000	3 5000	–	–
30	120 0000	265 0002	10 5119	205.3462	357 2737
31	155.0000	265.0002	10.5119	205 3462	357.2737
32	60 0000	272.5000	8.0000	–	–
33	143 5000	272 5000	5.5000	–	–
34	95 0000	280 0000	–	–	–
35	130 0000	280.0000	–	–	–
36	47 5000	282.5000	3.5000	–	–
37	72 5000	282.5000	3.5000	–	–
38	40 0000	290.0000	–	–	–
39	80.0000	290.0000	–	–	–
40	95.0000	290.0000	–	–	–
41	115 0000	290 0000	–	–	–
42	130.0000	290 0000	–	–	–
43	150 0000	290 0000	–	–	–
44	30 0000	300 0000	–	–	–
45	185.0000	300 0000	–	–	–

Table 7.5: List of vertices after lexicographical ordering.

Edge No	Entity type	Start vert No	End vert No
1	4	1	2
2	4	1	44
3	4	2	45
4	4	3	4
5	4	3	20
6	4	4	13
7	4	5	6
8	4	5	14
9	4	6	21
10	4	7	8
11	4	7	22
12	4	8	23
13	1	9	–
14	1	10	–
15	1	11	–
16	1	12	–
17	4	13	14
18	1	15	–
19	1	16	–
20	1	17	–
21	1	18	–
22	1	19	–
23	4	20	21
24	4	22	23
25	4	24	25
26	4	24	38
27	4	25	39
28	3	26	30
29	4	26	34
30	3	27	31
31	4	27	35
32	1	28	–
33	1	29	–
34	4	30	41
35	4	31	43
36	1	32	–
37	1	33	–
38	4	34	40
39	4	35	42
40	1	36	–
41	1	37	–
42	4	38	39
43	4	40	41
44	4	42	43
45	4	44	45

Table 7.6: List of edges after lexicographical ordering

part : Various components used in VMC-345
material : 1010 steel
units : mm
thickness : 2 00

Component/ layout type : 2-D

No. of features : 21

Vertices present in various features :

Feature No	Vertices Nos & their degree				
1	1 (2)	2 (2)	44 (2)	45 (2)	
2	3 (2)	4 (2)	5 (2)	6 (2)	13 (2)
	14 (2)	20 (2)	21 (2)		
3	7 (2)	8 (2)	22 (2)	23 (2)	
4	9 (0)				
5	10 (0)				
6	11 (0)				
7	12 (0)				
8	15 (0)				
9	16 (0)				
10	17 (0)				
11	18 (0)				
12	19 (0)				
13	24 (2)	25 (2)	38 (2)	39 (2)	
14	26 (2)	30 (2)	34 (2)	40 (2)	41 (2)
15	27 (2)	31 (2)	35 (2)	42 (2)	43 (2)
16	28 (0)				
17	29 (0)				
18	32 (0)				
19	33 (0)				
20	36 (0)				
21	37 (0)				

Raw material feature set (R) : $\{1\}$
Boundary feature set (B) : $\{2, 3, 13, 14, 15\}$
Inside feature set (N) : $\{4, \ , 12, 16, .. , 21\}$
Component set (C) : $\{1, 2, 3, 4, 5\}$
Inside features of a comp. : $c_1 = 1; c_1 = \{4, 5, 10, 11, 12\}$, $c_2 = 2; c_2 = \{6, 7, 8, 9\}$, $c_3 = 3; c_3 = \{16, 17, 18, 20, 21\}$, $c_4 = 4; c_4 =$, $c_5 = 5; c_5 = \{19\}$,

Pseudo vertices :

Vert No	x	y
30'	109 500	255 000
31'	144.500	255.000

MER of components :

Comp No	Lower left vertex		Upper right vertex	
	x	y	x	y
1	40 000	205 000	125 000	245 000
2	135 500	210 000	175 000	245 000
3	40 000	255 000	80 000	290 000
4	95 000	255 000	120 000	290 000
5	130 000	255 000	155 000	290 000

ACDR Table :

$c_{j'}$ / c_j	1	2	3	4	5
1	–	1^{5}	1^{2}	1^{5}	
2	1^{7}	–			1^{5}
3	1^{4}		–	1^{1}	
4	1^{7}		1^{3}	–	1^{1}
5		1^{7}		1^{3}	–

Raw material :

type . sheet
feature No. . 1
size · 155.000 105.000
position · 30 000 195.000 185 000 300 000

Component No. : 1

Inside Features

piercing : 1
feature No. · 4
position : 46.000 211.000 3 000

piercing : 2
feature No. : 5
position : 119.000 211 000 3 000

piercing : 3
feature No. : 10
position : 46 000 239.000 3.000

piercing : 4
feature No . 11
position . 82.000 239.000 3 000

piercing : 5
feature No. : 12
position : 119.000 239 000 3 000

Boundary Features

blanking :
shape — outline boundary of the component
feature no — 2
position — 40 000 205 000
40 000 245 000
125 000 245 500
125 000 245 500

notching : 1
position — 65 000 205 000
65 000 225 000
100 000 225 000
100 000 205 000

blanking with notching :
position — 40 000 205 000
40 000 245 000
125 000 245 000
125 000 205 000
100 000 205 000
100 000 225 000
65 000 225 000
65 000 205 000

Note : Piercing, blanking, notching, and blanking with notching are not given for other components, since the output list is very large

Outside Features

shearing : 1
direction — Y axis
range — 10 000
position — 30 000 195.000 40 000 195.000
30.000 300 000 40.000 300.000

shearing : 2
direction — Y axis
range — 5 000
position — 125 000 195 000 130.000 195 000
125 000 300.000 130 000 300.000

shearing : 3
direction — Y axis
range : 10 000
position : 175.000 195.000 185.000 195.000
175.000 300 000 185 000 300.000

Example-3

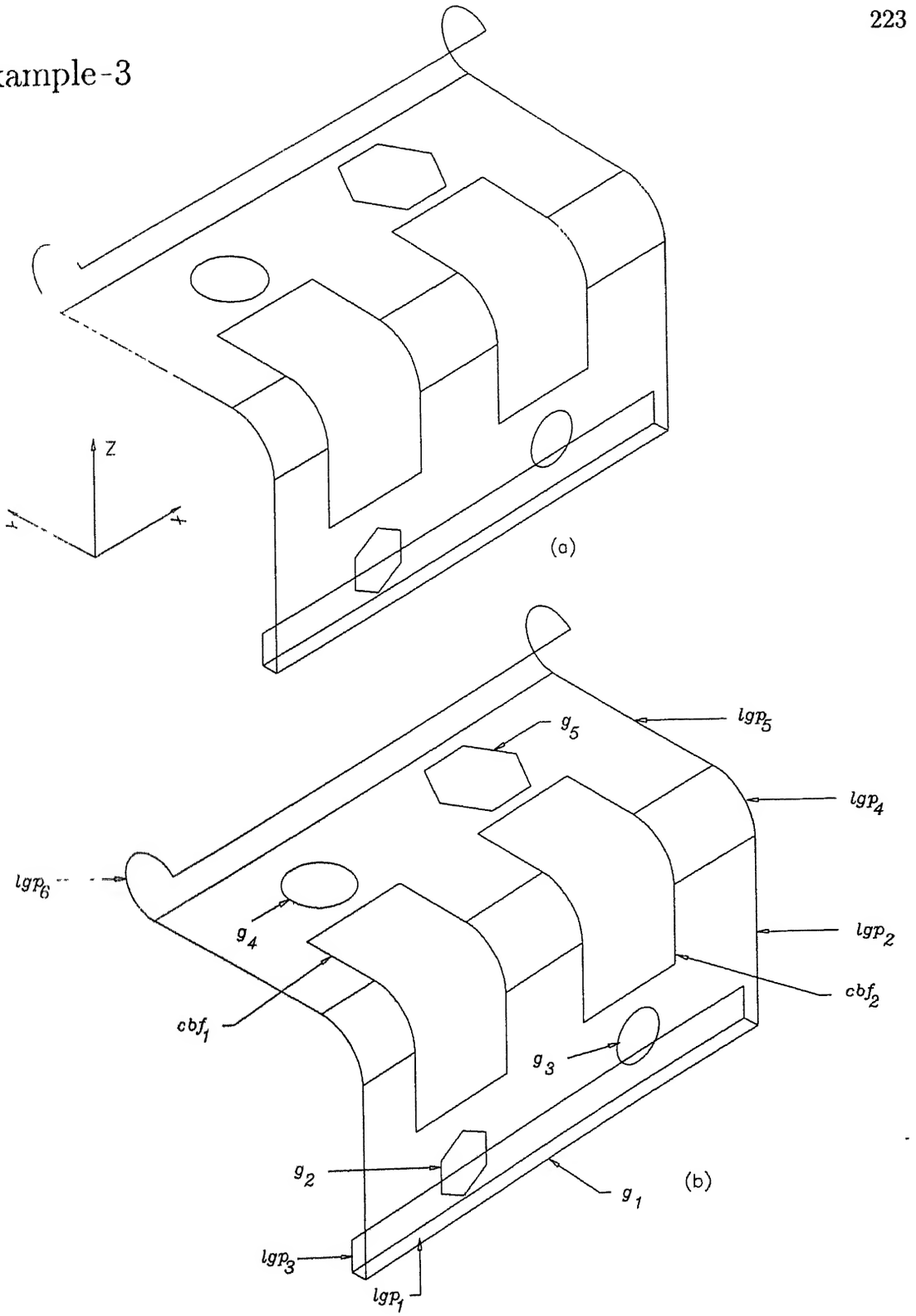


Figure 7.9: A 3–D component (Example–3) (a) Drawing of the component given as input to the system, and (b) Subgraph and plane numbers are represented for easy reference

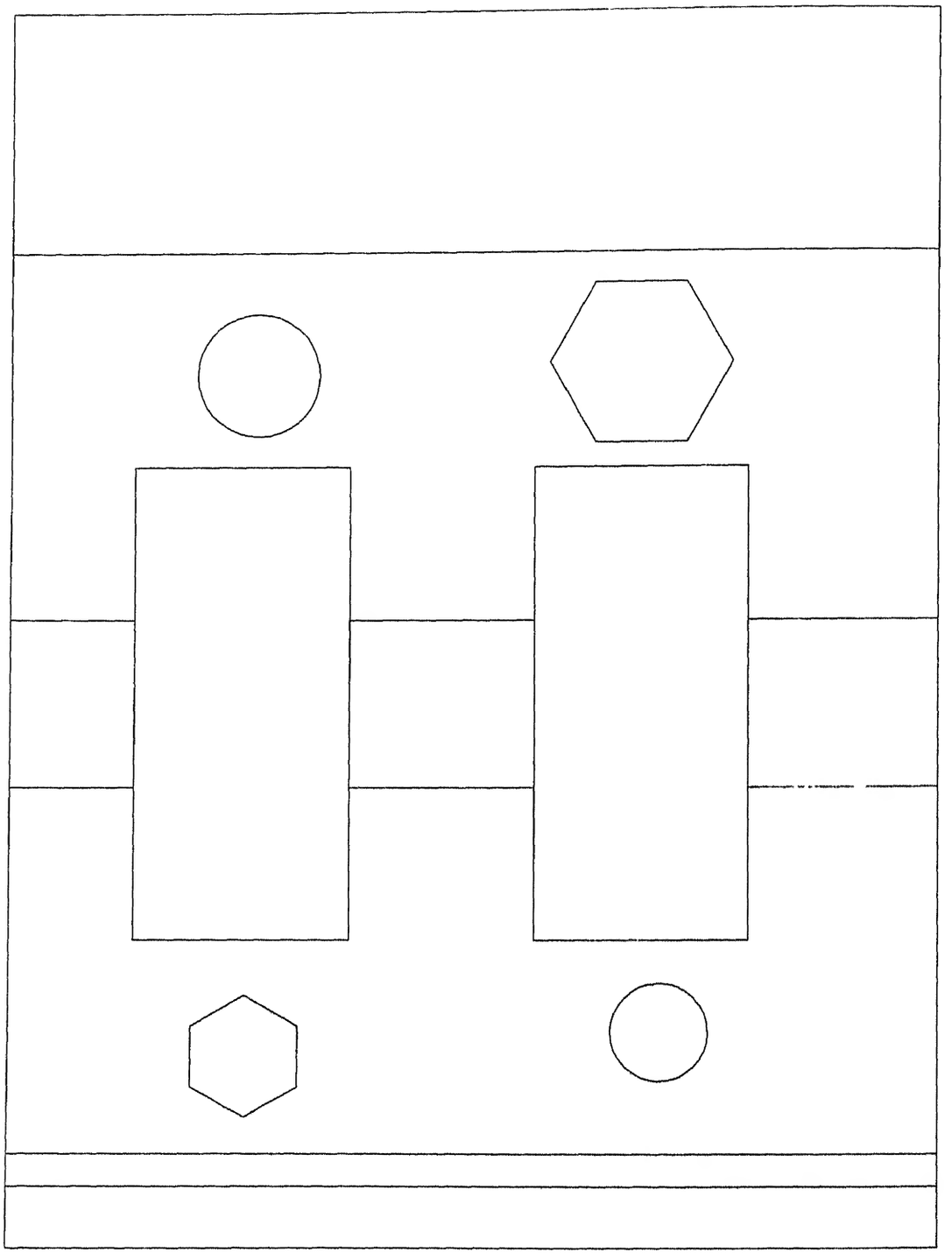


Figure 7.10: Flat pattern development of the 3–D component shown in Fig. 7.9.

Vert No.	x	y	z	rad /bul	start ang	end ang
1	201 7701	108 7299	-32 2200	–	–	–
2	351 7701	108 7299	-32 2200	–	–	–
3	201 7701	113 7299	-32 2200	–	–	–
4	351 7701	113 7299	-32 2200	–	–	–
5	239.7701	108 7299	-26 2200	–	–	–
6	201.7701	113 7299	-22 2200	–	–	–
7	351 7701	113 7299	-22.2200	–	–	–
8	231 1098	108 7299	-21 2200	–	–	–
9	248 4303	108 7299	-21 2200	–	–	–
10	307 2701	108 7299	-12 2200	8 0000	–	–
11	231.1098	108.7299	-11 2200	–	–	–
12	248 4303	108 7299	-11 2200	–	–	–
13	239 7701	108 7299	-6 2200	–	–	–
14	221 7701	108 7299	2 7800	–	–	–
15	256.7701	108.7299	2 7800	–	–	–
16	286 7701	108.7299	2 7800	–	–	–
17	321.7701	108 7299	2 7800	–	–	–
18	201.7701	108 7299	27 7800	15 0000	0 0000	90 0000
19	221.7701	108.7299	27.7802	15 0000	0 0000	90 0000
20	256.7701	108 7299	27 7800	15 0000	0 0000	90 0000
21	286.7701	108.7299	27 7802	15 0000	0 0000	90 0000
22	321 7701	108 7299	27 7800	15 0000	0 0000	90 0000
23	351.7701	108 7299	27 7800	15 0000	0 0000	90 0000
24	201.7701	123.7299	42.7800	–	–	–
25	221.7701	123.7299	42 7800	–	–	–
26	256.7701	123.7299	42 7800	–	–	–
27	286.7701	123.7299	42 7800	–	–	–
28	321.7701	123 7299	42 7800	–	–	–
29	351.7701	123.7299	42 7800	–	–	–
30	221 7701	148.7299	42 7800	–	–	–
31	256.7701	148.7299	42 7800	–	–	–
32	286.7701	148.7299	42.7800	–	–	–
33	321 7701	148.7299	42 7800	–	–	–
34	296.7701	152.7492	42 7800	–	–	–
35	311.7701	152.7492	42 7800	–	–	–
36	241.7701	163.7299	42 7800	10 0000	–	–
37	289.2701	165.7395	42 7800	–	–	–
38	319 2701	165.7395	42 7800	–	–	–
39	296.7701	178.7299	42 7800	–	–	–
40	311.7701	178 7299	42 7800	–	–	–
41	201.7701	183.7299	42.7800	10 0000	45 0000	270 0000
42	351.7701	183.7299	42 7800	10 0000	45 0000	270 0000
43	201.7701	176.6589	59 8510	–	–	–
44	351.7701	176.6588	59 8511	–	–	–

Table 7.7: List of vertices after lexicographical ordering

Edge No	Entity type	Start vert No	End vert No
1	4	1	2
2	4	1	3
3	4	1	18
4	4	2	4
5	4	2	23
6	4	3	4
7	4	3	6
8	4	4	7
9	4	5	8
10	4	5	9
11	4	6	7
12	4	8	11
13	4	9	12
14	1	10	
15	4	11	13
16	4	12	13
17	4	14	15
18	4	14	19
19	4	15	20
20	4	16	17
21	4	16	21
22	4	17	22
23	4	18	19
24	3	18	24
25	3	19	25
26	4	20	21
27	3	20	26
28	3	21	27
29	4	22	23
30	3	22	28
31	3	23	29
32	4	24	25
33	4	24	41
34	4	25	30
35	4	26	27
36	4	26	31
37	4	27	32
38	4	28	29
39	4	28	33
40	4	29	42
41	4	30	31
42	4	32	33
43	4	34	35
44	4	34	37
45	4	35	38
46	1	36	
47	4	37	39
48	4	38	40
49	4	39	40
50	4	41	42
51	3	41	43
52	3	42	44
53	4	43	44

Table 7.8: List of edges after lexicographical ordering.

part : Simple 3-D component
material : steel
units : mm
thickness : 2 00

Component/ layout type : 3-D

No. of subgraphs : 5

Vertices present in various subgraphs :

Graph No	Vertices Nos & their degree					
1	1 (3)	2 (3)	3 (3)	4 (3)	6 (2)	7 (2)
	14 (2)	15 (2)	16 (2)	17 (2)	18 (3)	19 (3)
	20 (3)	21 (3)	22 (3)	23 (3)	24 (3)	25 (3)
	26 (3)	27 (3)	28 (3)	29 (3)	30 (2)	31 (2)
	32 (2)	33 (2)	41 (3)	42 (3)	43 (2)	44 (2)
2	5 (2)	8 (2)	9 (2)	11 (2)	12 (2)	13 (2)
3	10 (0)					
4	34 (2)	35 (2)	37 (2)	38 (2)	39 (2)	40 (2)
5	36 (0)					

Parent subgraph (G_p) : {1}
Number of planes in G_p : 6

Child subgraph (G_c) : {2, 3, 4, 5}

Parent subgraph

Planes and edges present :

Plane No	Edge Nos present in a plane					
1	1	2	4	6		
2	1	3	5	17	18	19
	20	21	22	23	26	29
3	6	7	8	11		
4	23	24	25	26	27	28
	29	30	31	32	35	38
5	32	33	34	35	36	37
	38	39	40	41	42	50
6	50	51	52	53		

Curved planes (bp_1) :

curved plane No. : 1
plane No : 4

curved plane No. : 2
plane No : 6

APR Table :

$l_{1,pl'}$ / $l_{1,pl}$	1	2	3	4	5	6
1	–	1	1			
2	1	–		1		
3	1		–			
4		1		–	1	
5				1	–	1
6					1	–

Cross bend features (cbf_1) :

cross bend feature No. : 1
edges : 17 18 19 25 27 34 36 41

cross bend feature No. : 2
edges : 20 21 22 28 30 37 39 42

Child subgraphs

child subgraph No. : 1
type : 2–D
present in the plane of G_p : 2
edges present : 9 10 12 13 15 16

child subgraph No. : 2
type : 2–D
present in the plane of G_p : 2
edges present : 14

child subgraph No. : 3
type : 2–D
present in the plane of G_p : 5
edges present : 43 44 45 47 48 49

child subgraph No. : 4
type : 2–D
present in the plane of G_p : 5
edges present : 46

Flat pattern development vertices list :

Vert No	x	y	z	rad /bul	start ang	end ang
1	18.0000	104 0000	0 0000	–	–	–
2	168 0000	104 0000	0 0000	–	–	–
3	18 0000	114 0000	0 0000	–	–	–
4	168 0000	114 0000	0 0000	–	–	–
5	18 0000	119 2624	0 0000	–	–	–
6	168 0000	119 2624	0.0000	–	–	–
7	56 0000	125 2624	0 0000	–	–	–
8	47 2603	130 2624	0.0000	–	–	–
9	64 5809	130 2624	0 0000	–	–	–
10	123 4206	139 2624	0 0000	8 0000	–	–
11	47 2603	140 2624	0.0000	–	–	–
12	64 5809	140 2624	0 0000	–	–	–
13	56 0000	145 2624	0 0000	–	–	–
14	38 0000	154 2624	0 0000	–	–	–
15	73 0000	154 2624	0.0000	–	–	–
16	103 0000	154 2624	0 0000	–	–	–
17	138 0000	154 2624	0 0000	–	–	–
18	18 0000	179 2624	0 0000	–	–	–
19	38 0000	179 2626	0 0000	–	–	–
20	73 0000	179 2624	0 0000	–	–	–
21	103 0000	179.2626	0 0000	–	–	–
22	138 0000	179 2624	0 0000	–	–	–
23	168 0000	179 2624	0 0000	–	–	–
24	18 0000	206 6247	0 0000	–	–	–
25	38 0000	206 6246	0 0000	–	–	–
26	73 0000	206 6247	0 0000	–	–	–
27	103 0000	206 6246	0 0000	–	–	–
28	138 0000	206 6246	0 0000	–	–	–
29	168 0000	206.6247	0 0000	–	–	–
30	38 0000	231 6246	0 0000	–	–	–
31	73 0000	231 6246	0 0000	–	–	–
32	103.0000	231 6246	0 0000	–	–	–
33	138 0000	231.6246	0 0000	–	–	–
34	113 0000	235 6246	0 0000	–	–	–
35	128 0000	235 6246	0.0000	–	–	–
36	58 0000	246 6246	0.0000	10 0000	–	–
37	105.4206	248 6246	0 0000	–	–	–
38	135 4206	248 6246	0 0000	–	–	–
39	113 0000	261.6246	0 0000	–	–	–
40	128 0000	261 6246	0 0000	–	–	–
41	18 0000	266 6246	0 0000	–	–	–
42	168 0000	266 6246	0 0000	–	–	–
43	18 0000	305 8846	0 0000	–	–	–
44	168 0000	305 8846	0 0000	–	–	–

Flat pattern development : Figure 7 10

Single plane features of the parent subgraph

curvature bend	:	1
plane No		4
curling	:	1
plane No		6

Multi–plane features of the parent subgraph

hemming	:	1
plane Nos		1 2 3

Shearing features

Parent subgraph :

piercing	:	1
cross bend feature No.		1
piercing	:	2
cross bend feature No		2

Child subgraph :

piercing	:	3
subgraph No.		2
piercing	:	4
subgraph No		3
piercing	:	5
subgraph No		3
piercing	:	6
subgraph No		4
piercing	:	7
subgraph No		5

Example–4

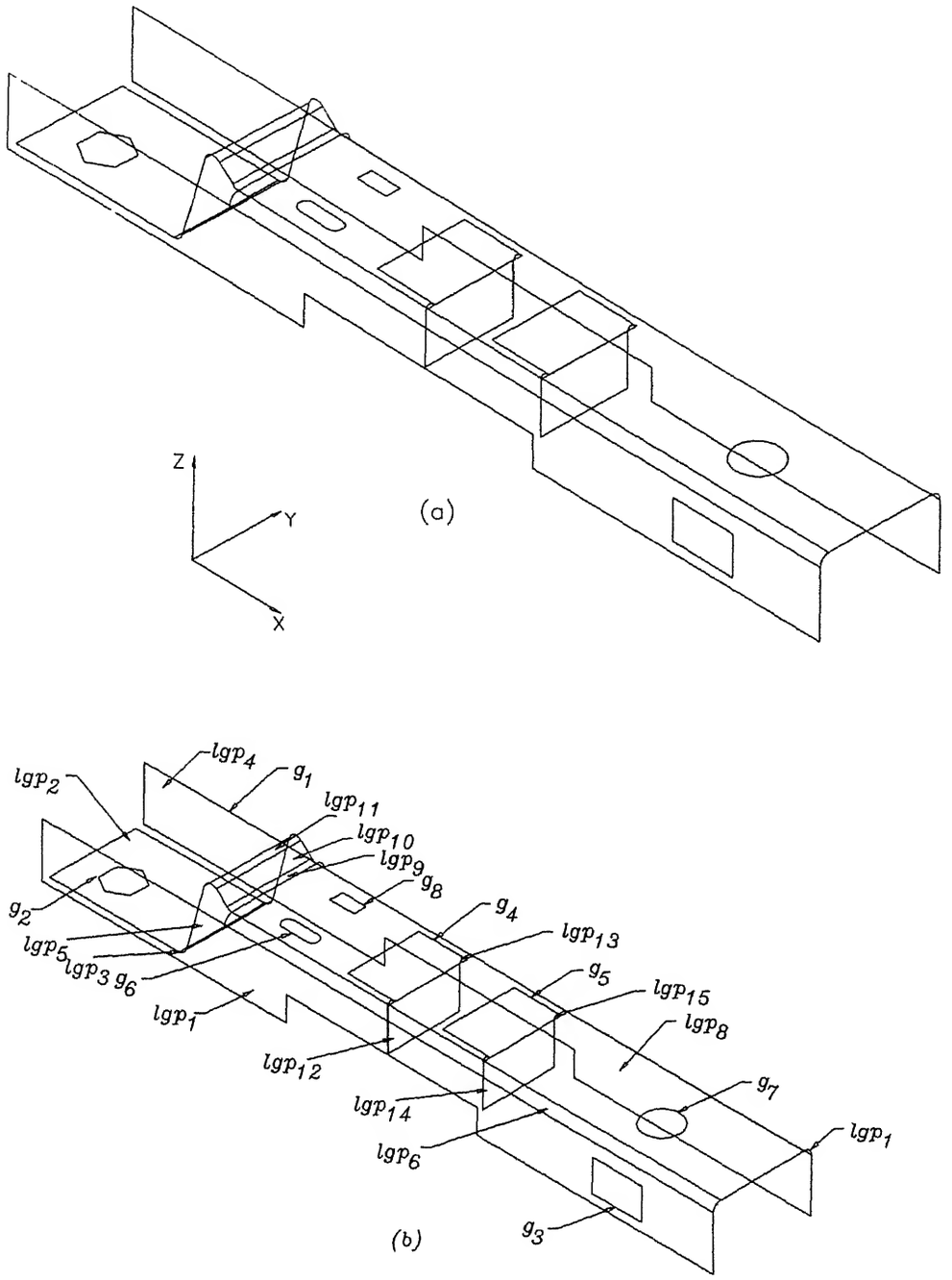


Figure 7.11: A complex 3–D component (a) Drawing of a modified part of sun microphone box given as input to the system, and (b) Representation of graph and plane numbers for easy reference.

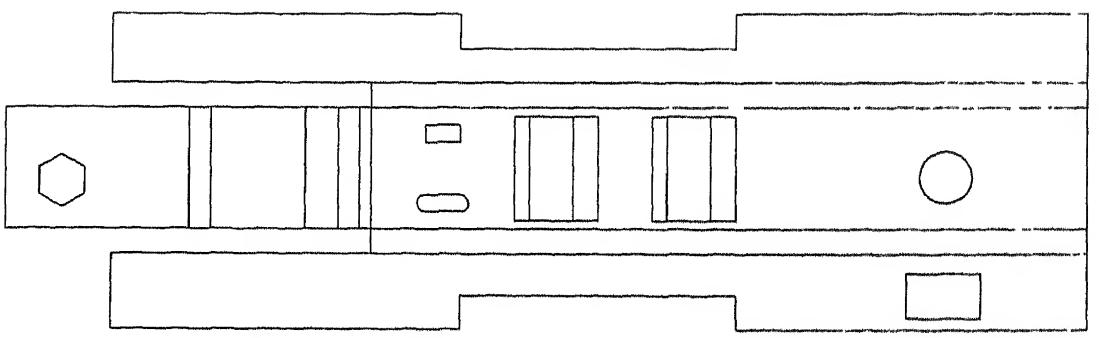


Figure 7.12: Flat pattern development of the 3–D component shown in Fig. 7.11.

Vert No	x	y	z	rad /bul	start ang	end ang
1	-101 3144	101 3144	143 2802	–	–	–
2	-1 3144	101 3144	143 2802	–	–	–
3	78 6856	101 3144	143 2802	–	–	–
4	178 6856	101 3144	143 2802	–	–	–
5	-101.3144	104 3144	143 2802	–	–	–
6	-48 5106	104 3144	143 2802	3 0000	270 0000	345 0000
7	-84 8192	110 5644	143 2802	–	–	–
8	-91 3144	114 3144	143 2802	–	–	–
9	-78 3240	114 3144	143 2802	–	–	–
10	-91.3144	121 8144	143 2802	–	–	–
11	-78 3240	121 8144	143 2802	–	–	–
12	-84 8192	125 5644	143 2802	–	–	–
13	-101 3144	139 3144	143 2802	–	–	–
14	-48.5106	139 3144	143 2802	3 0000	270 0000	345.0000
15	-101.3144	142 3144	143 2802	–	–	–
16	-1 3144	142.3144	143 2802	–	–	–
17	78 6856	142.3144	143 2802	–	–	–
18	178.6856	142 3144	143.2802	–	–	–
19	-45.6128	104 3144	145 5037	–	–	–
20	-45.6128	139 3144	145 5037	–	–	–
21	127 5133	101 3144	146 7729	–	–	–
22	148.5106	101 3144	146 7729	–	–	–
23	35.6856	106 8144	148 2802	–	–	–
24	75.6856	106 8144	148.2802	–	–	–
25	35 6856	136.8144	148 2802	–	–	–
26	75 6856	136 8144	148.2802	–	–	–
27	-1.3144	101.3144	153.2802	–	–	–
28	78.6856	101 3144	153 2802	–	–	–
29	-1.3144	142.3144	153.2802	–	–	–
30	78.6856	142 3144	153 2802	–	–	–
31	127.5133	101 3144	159 8154	–	–	–
32	148 5106	101.3144	159 8154	–	–	–
33	-101.3144	101 3144	165.2802	–	–	–

Table 7.9: List of vertices after lexicographical ordering.

Vert No	x	y	z	rad /bul	start ang	end ang
34	-27 1612	101 3144	165 2802	3 0000	90 0000	180 0000
35	178 6856	101 3144	165 2802	3 0000	90 0000	180 0000
36	35 6856	106 8144	165 2802	3 0000	90 0000	180 0000
37	75 6856	106 8144	165 2802	3 0000	90 0000	180.0000
38	35 6856	136 8144	165 2802	3 0000	90.0000	180 0000
39	75 6856	136 8144	165 2802	3 0000	90 0000	180 0000
40	-101 3144	142 3144	165 2802	–	–	–
41	-27 1612	142 3144	165 2802	3 0000	0.0000	90 0000
42	178 6856	142 3144	165 2802	3.0000	90 0000	180 0000
43	-27 1612	104 3144	168 2802	3 0000	228.0000	270 0000
44	178 6856	104 3144	168 2802	–	–	–
45	18 6856	106 8144	168 2802	–	–	–
46	38 6856	106 8144	168 2802	–	–	–
47	58 6856	106 8144	168 2802	–	–	–
48	78 6856	106 8144	168.2802	–	–	–
49	-11 3144	109 3144	168 2802	2 5000	90.0000	270.0000
50	-1 3144	109.3144	168 2802	2.5000	270 0000	90 0000
51	-11 3144	114 3144	168 2802	–	–	–
52	-1 3144	114 3144	168.2802	–	–	–
53	138 6856	119 3144	168 2802	7 5000	–	–
54	-11 3144	129 3144	168 2802	–	–	–
55	-1 3144	129 3144	168.2802	–	–	–
56	-11 3144	134 3144	168.2802	–	–	–
57	-1 3144	134 3144	168 2802	–	–	–
58	18 6856	136 8144	168 2802	–	–	–
59	38 6856	136 8144	168 2802	–	–	–
60	58 6856	136.8144	168.2802	–	–	–
61	78 6856	136.8144	168.2802	–	–	–
62	-27.1612	139.3144	168 2802	3.0000	228.0000	270.0000
63	178.6856	139.3144	168.2802	–	–	–
64	-29.1706	104.3144	169 0526	3.0000	228.0000	270 0000
65	-29 1706	139 3144	169.0526	–	–	–
66	-38 5989	104 3144	171.6798	3 0000	228.0000	270.0000
67	-38.5989	139 3144	171 6798	3.0000	228.0000	270.0000
68	-33 6917	104.3144	173.1309	–	–	–
69	-33 6917	139 3144	173.1309	–	–	–

Table 7.9: List of vertices after lexicographical ordering (continued).

Edge No	Entity type	Start vert No	End vert No
1	4	1	2
2	4	1	33
3	4	2	27
4	4	3	4
5	4	3	28
6	4	4	35
7	4	5	6
8	4	5	13
9	4	6	14
10	3	6	19
11	4	7	8
12	4	7	9
13	4	8	10
14	4	9	11
15	4	10	12
16	4	11	12
17	4	13	14
18	3	14	20
19	4	15	16
20	4	15	40
21	4	16	29
22	4	17	18
23	4	17	30
24	4	18	42
25	4	19	20
26	4	19	66
27	4	20	67
28	4	21	22
29	4	21	31
30	4	22	32
31	4	23	25
32	4	23	36
33	4	24	26
34	4	24	37
35	4	25	38
36	4	26	39
37	4	27	28
38	4	29	30
39	4	31	32
40	4	33	34

Table 7.10: List of edges after lexicographical ordering

Edge No	Entity type	Start vert No	End vert No
41	4	34	35
42	3	34	43
43	3	35	44
44	4	36	38
45	3	36	46
46	4	37	39
47	3	37	48
48	3	38	59
49	3	39	61
50	4	40	41
51	4	41	42
52	3	41	62
53	3	42	63
54	4	43	44
55	4	43	62
56	3	43	64
57	4	44	63
58	4	45	46
59	4	45	58
60	4	46	59
61	4	47	48
62	4	47	60
63	4	48	61
64	4	49	50
65	3	49	51
66	3	50	52
67	4	51	52
68	1	53	-
69	4	54	55
70	4	54	56
71	4	55	57
72	4	56	57
73	4	58	59
74	4	60	61
75	4	62	63
76	3	62	65
77	4	64	65
78	4	64	68
79	4	65	69
80	4	66	67
81	3	66	68
82	3	67	69
83	4	68	69

Table 7.10: List of edges after lexicographical ordering (continued).

part : Modified complex part of a microphone box
material : Aluminum (assumed)
units : mm
thickness : 1 00

Component/ layout type : 3-D

No. of subgraphs : 8

Vertices present in various subgraphs :

Graph No	Vertex No and its degree					
1	1 (2)	2 (2)	3 (2)	4 (2)	5 (2)	6 (3)
	13 (2)	14 (3)	15 (2)	16 (2)	17 (2)	18 (2)
	19 (3)	20 (3)	27 (2)	28 (2)	29 (2)	30 (2)
	33 (2)	34 (3)	35 (3)	40 (2)	41 (3)	42 (3)
	43 (4)	44 (3)	62 (4)	63 (3)	64 (3)	65 (3)
	66 (3)	67 (3)	68 (3)	69 (3)		
2	7 (2)	8 (2)	9 (2)	10 (2)	11 (2)	12 (2)
3	21 (2)	22 (2)	31 (2)	32 (2)		
4	23 (2)	25 (2)	36 (3)	38 (3)	45 (2)	46 (3)
	58 (2)	59 (3)				
5	24 (2)	26 (2)	37 (3)	39 (3)	47 (2)	48 (3)
	60 (2)	61 (3)				
6	49 (2)	50 (2)	51 (2)	52 (2)		
7	53 (0)					
8	54 (2)	55 (2)	56 (2)	57 (2)		

Parent subgraph (G_p) : {1}
Number of planes in G_p : 11

Child subgraph (G_c) : {2, 3, 4, 5 6, 7, 8}

Parent subgraph

Planes and edges present :

Plane No.	Edges No present in a plane				
1	1	2	3	4	5
	6	37	40	41	
2	7	8	9	17	
3	9	10	18	25	
4	19	20	21	22	23
	24	38	50	51	
5	25	26	27	80	
6	41	42	43	54	
7	51	52	53	75	
8	54	55	57	75	
9	55	56	76	77	
10	77	78	79	83	
11	80	81	82	83	

Curved planes (bp_1) :

curved plane No. : 1
plane No : 3

curved plane No. : 2
plane No : 6

curved plane No. : 3
plane No : 7

curved plane No. : 4
plane No : 9

curved plane No. : 5
plane No : 11

APR Table :

$l_{1,pl'}$ / $l_{1,pl}$	1	2	3	4	5	6	7	8	9	10	11
1	–					1					
2		–	1								
3		1	–		1						
4				–			1				
5			1		–						1
6	1					–		1			
7				1			–	1			
8						1	1	–	1		
9								1	–	1	
10									1	–	1
11					1					1	–

Child subgraphs

Child subgraph No. : 1
graph No 2
type 2–D
present in the plane of G_p : 2
edges present : 11 12 13 14 15 16

Child subgraph No. : 2
graph No 3
type : 2–D
present in the plane of G_p 1
edges present 28 29 30 39

Child subgraph No. : 3
graph No. : 4
type : 3–D

present in the plane of G_p : 8

edges present : 58 59 60 73
number of planes : 2

Plane No.	Edges present in the planes
12	31 32 35 44
13	44 45 48 60

Plane No	Adjacent planes
12	13
13	12

Child subgraph No. : 4
graph No : 5
type : 3-D
present in the plane of G_p : 8
edges present : 61 62 63 74
number of planes : 2

Plane No	Edges present in the planes
14	33 34 36 46
15	46 47 49 63

Plane No.	Adjacent planes
14	15
15	14

Child subgraph No. : 5
graph No. : 6
type : 2-D
present in the plane of G_p : 8
edges present : 64 65 66 67

Child subgraph No. : 6
graph No. : 7
type : 2-D
present in the plane of G_p : 8
edges present : 68

Child subgraph No. : 7
graph No : 8
type : 2-D
present in the plane of G_p : 8
edges present : 69 70 71 72

Flat pattern development : Figure 7.12

Single plane features of the parent subgraph

curvature bend : 1
plane No. : 3

curvature bend : 2

plane No 6

curvature bend : 3
plane No 7

curvature bend : 4
plane No 9

curvature bend : 5
plane No 11

Multi–plane of the parent–child subgraph features

louvering : 1
child subgraph No 4

louvering : 2
child subgraph No 5

Shearing features

Parent subgraph :

notching : 1
position 117 3582 209 9451
117 3582 199 9451
197 3582 199.9451
197 3582 209 9451

notching : 2
position 197 3582 118 3625
197.3582 128 3625
117 3582 118 3625
117.3582 128 3625

Child subgraph :

piercing : 1
subgraph No 2

piercing : 2
subgraph No : 3

piercing : 3
subgraph No . 6

piercing : 4
subgraph No · 7

piercing : 5
subgraph No. . 8

lancing : 1
subgraph No · 4
edge No. . 58 59 73

lancing	:	2		
subgraph No		5		
edge No		61	62	74

Chapter 8

Conclusions and Scope for Future Research

8.1 Conclusions

In the chapter on Introduction and Literature Review, different approaches used for representing a physical entity in the computer environment in general and for sheet metal components in particular were discussed It has been observed that a 3–D solid model is converted into face oriented structure, foil type, or some other form, such that the thickness of the sheet is neglected for identifying the features. As such, in the present research work wireframe model has been used for recognizing features. This geometric model is considered as an appropriate choice because of its extensive use in sheet metal industries, and the advantages it offers with respect to hardware and software Pressworking features are recognized from the plan view in case of a 2-D part nested layout, and the mean plane of the component in case of a 3-D sheet metal component The thickness of the sheet metal is given to the system interactively. Set theoretic and graph based approach are used to identify shearing and forming operations by developing a set of principles and relating them geometrically and topologically. This approach helps the designer to create a component flexibly and without any hindrance since the entities used in the wireframe model are the same as those used in manual practices to depict drawings of an object. The approach helps in identification of all the features in a hierarchical way from the CAD database. The identified features represent the potential candidates for manufacturing the component. Thus, the proposed approach makes the CAD database file of a wireframe model meaningful to the CAM environment and as such can be considered as *a step towards the realization of CIMS in true sense.*

In this research, classification systems for entity groups, pressworking operations and features are proposed, and they are :

- classification system for entity groups

- classification system for shearing operations and features
- classification system for forming operations and features

Further, a set of principles and generalized algorithms for recognizing pressworking features from wireframe model are proposed They are

- An algorithm for developing entity groups from the flexibly represented component/part layout adhering to the Euler's formula for wireframe model, and principles for identifying the nature of a component (i.e. 2-D or 3-D).
- A generalized algorithm for the ray containment test to uniquely identify the presence of an entity group within another entity group by taking the various special cases of the ray crossings with respect to the boundary feature. The generalized algorithm also used to extract the component set from the feature set.
- An algorithm based on the generalized principles for identifying the raw material type from a 2-D component layout using the geometry and topology of the raw material feature from the CAD database of wireframe model.
- A set of principles for identifying various shearing and forming features and to characterize them by relating their geometry and topology.
- A set of generalized principles for recognizing parent subgraph and child subgraphs from a 3-D component graph using the topological concepts of the component
- The principle of vector normal method for identifying the various loops present in the parent and child subgraphs. Also, the loop of the child subgraph present in the loop of the parent subgraph is identified.
- An algorithm for extracting and identifying cross bend features from the component.
- An algorithm for converting the non-Euler representation of the flat pattern developed for a 3-D component to the form which satisfies the Euler's formula for wireframe model. Shearing features are then identified from the nested layout created by the designer in AutoCAD with the help of the flat pattern of developed for a 3-D component.

Thus the proposed methodology is of generalized nature for identifying pressworking features from 2–D and 3–D components created by wireframe model irrespective of the CAD database. Further, the methodology developed is not dependent on the data structure used in this research work.

The proposed classification systems help in determining features in a hierarchical manner and thus reduce the search requirements during implementation. The classification systems also help in easy identification of finish operations, if total tolerances are taken into consideration.

The proposed methodology generates information about different operations by means of which a part can be produced from raw material, and thus enables in the development of alternative process plans for manufacturing the part An appropriate process plan can be selected from the generated alternative process plans based upon the requirements and facilities available in an organization The geometrical information of identified features obtained from the system can be used for optimization of tool paths in case of CNC turret presses, off-line generation of G and M codes, generation of GT codes, mapping form features for the development of tools to produce the shape and for inspection of tools and parts etc.

8.2 Scope for future research

In this thesis, various pressworking features are identified from the CAD database (AutoCAD) of wireframe geometric modeler used to represent sheet metal components. As stated earlier, pressworking features are recognized with the help of a set of principles developed by relating their geometry and topology. Graph based approach is used for the generation of component graph, and for identifying subgraphs in it. Set theoretic concepts are used to identify various feature sets and types of subgraphs. No research work is complete in itself and is true for the present work also. For making the proposed system more complete, versatile and exhaustive, further research on the following suggested lines is required.

- A study to identify other entities (like ellipse, 3-D polyline, parabola, etc.) from the CAD database of wireframe modeler that can be used for the representation of sheet metal component/part layout.

- In this work, it has been assumed that the minimum raw material required for manufacturing the component is a rectangular shape (MER), with sides parallel to X and Y axes In practice, this may not always be true, since MER can be oriented in any direction with respect to the X and Y axes By relaxing this assumption, it will be possible to account for a more realistic environment.

- In case of 2-D components, for minimizing waste, a part that can be manufactured from the pierced material is manufactured first, then the outer component. Thus, a

component can be located totally inside another component. These type of layouts have to be considered for further identification of pressworking features.

- By combining similar and dissimilar features, it is possible to reduce the total number of features, and hence number of operations required to manufacture the component. This has to be done by checking the feasibility of the combination of features. The factors that have to be considered are design of tools, manufacturability, specifications of presses, etc Similarly, by considering the tools and presses available, the decomposition of features may also be required to manufacture the components.

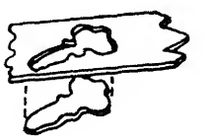
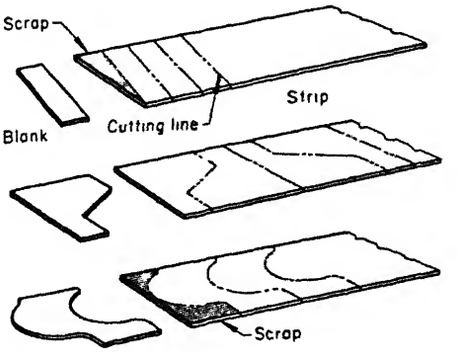
- Classification systems of forming operations and features are given in the present work, but few operations (deep drawing, embossing, etc.) are not yet characterized and recognized.

Extension of the present work based on the above suggested directions would enhance its utility and act as a step forward in bringing realistic CAD and CAM integration.

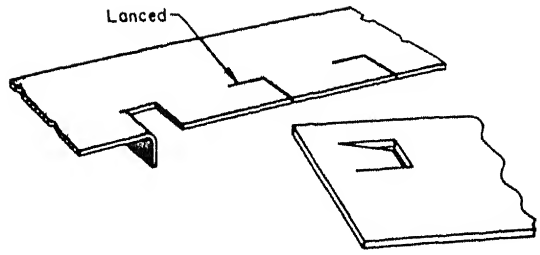
Appendix A

Definitions of Shearing Operations

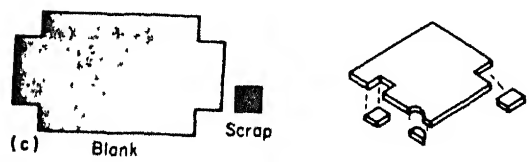
In this Appendix, definitions of various shearing operations are given (Metals Hand Book, Vol 4, 1970; Lascoe, 1988).

DEFINITIONS	FIGURES
Blanking : Blanking is a process in which, the complete outline of a workpiece is cut in a single press stroke from sheet or strip stock	
Cutoff : A cutoff operation cuts along a line to produce blanks without generating any scrap in the cutting operation This operation is performed after most of the part outline have been developed by notching or lancing operations in the preceding stations.	

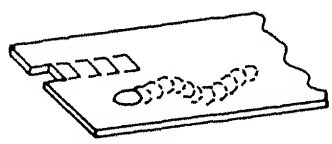
Lancing : Lancing is an operation in which a single line cut or slit is made part way across the strip stock, without removing any metal Generally it is performed to free metal for forming.



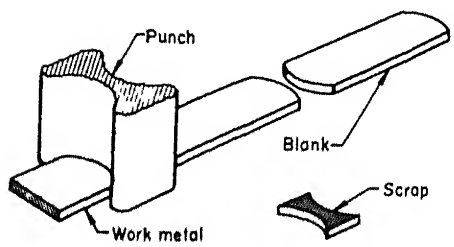
Notching : Notching is a process in which the individual punch removes a piece of metal from the edge of the blank



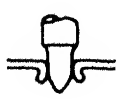
Nibbling : Nibbling is a progressive notching at high rate of speed, making either a smooth finished edge or a scalloped edge.



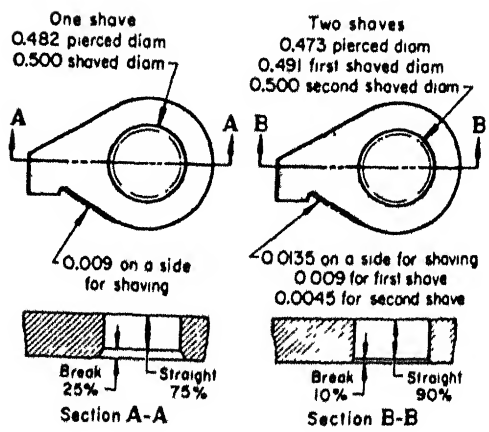
Parting : Parting separates blanks from parent material by cutting away a strip of material between them. Like cutoff, it is done after most of the outline of the part has been developed by notching or lancing.



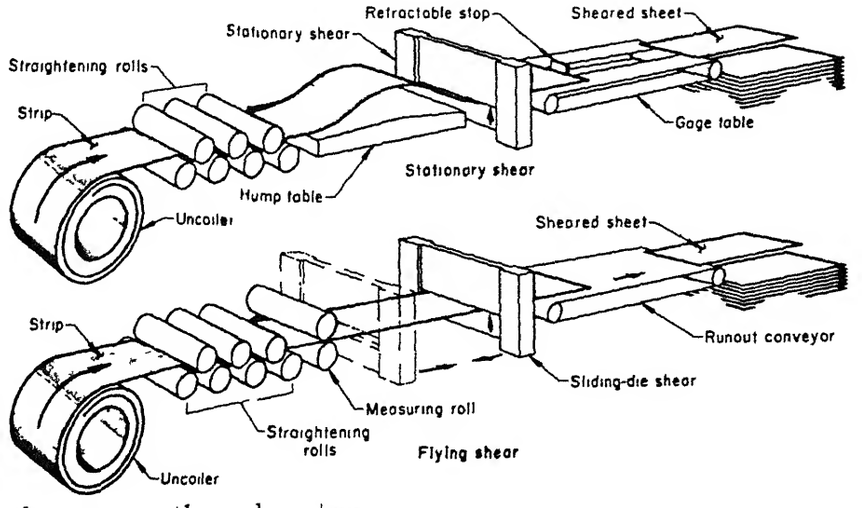
Piercing : Piercing is also known as punching, or stamping. This is similar to blanking except that the pierced out slug is the waste and the surrounding metal is the workpiece.



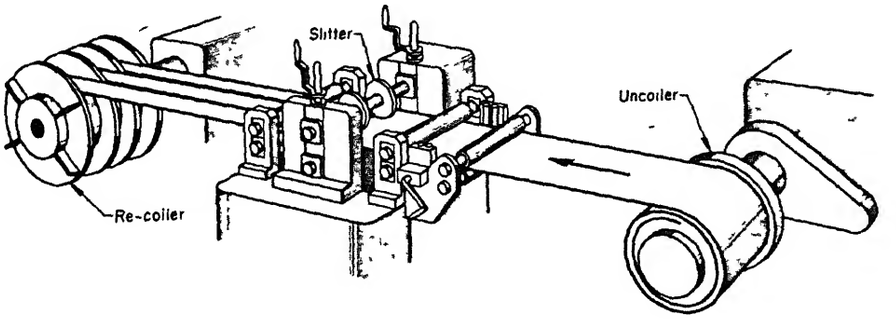
Shaving : Shaving is an operation used after blanking to give a smooth, square edge, and greater accuracy than can be achieved in an ordinary blanking.



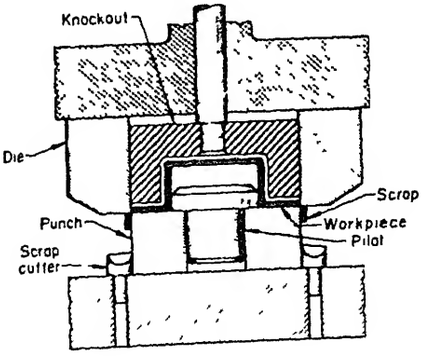
Shearing : Shearing is a process in which the material is cut with dies or blades In case of a coil or strip, it is used to cut the material to get specified length of the stock.



Slitting : In this process, the coil or strip is cut into narrower coils or strips for specified widths.



Trimming : Trimming is a process for removing excess material, such as deformed and uneven metal on drawn or formed parts, and metal that were needed in previous operations, like blank-holding flange for drawing operation



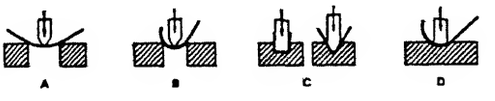
Appendix B

Definitions of Forming Operations

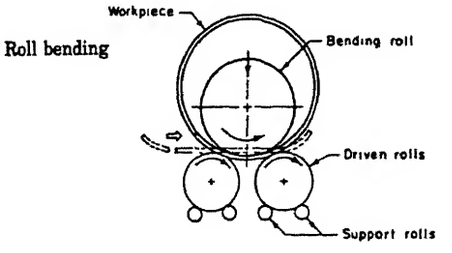
In this Appendix, definitions of various forming operations are given (Sachs, 1966, Metals Hand Book, Vol 4, 1970; Lascoe, 1988, Altan et al., 1983).

DEFINITIONS	FIGURES

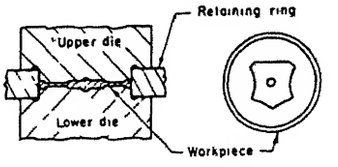
Bending : Bending is a process used for forming flat sheets into linear sections, such as angles, channels, hats, etc. There are three types of bending methods encountered, viz., air bending, die bending, and roll bending.



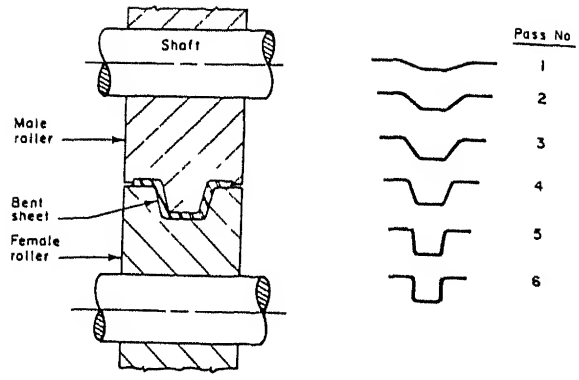
(A) air bending, (B) air rounding, (C) die bending, (D) die rounding



Coining : Coining is a closed die forming operation, usually performed cold, in which all surfaces of the work are confined or restrained, resulting in a well defined imprint of the die on the workpiece It is also a restricting operation used to sharpen or change an existing radius or profile.



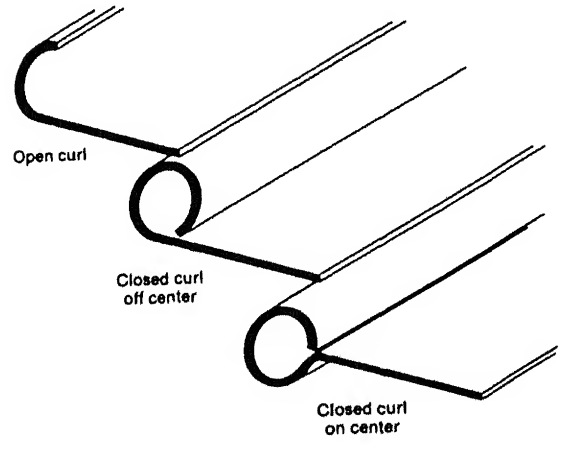
Contour roll forming : This process is used to produce long components having desired shapes of uniform cross section by feeding the stock longitudinally through a series of roll sections equipped with contoured rolls



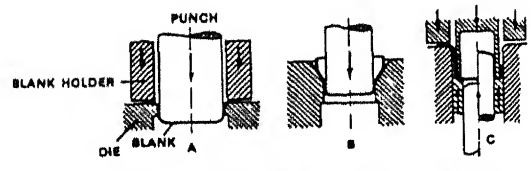
(a) Roller profiles for one pass

(b) Various passes for roll forming "HAT" section

Curling : The process is used for forming an edge of circular cross section along a sheet or along the end of a shell or tube, either to inside or outside of it. Different types of curling are open, closed, and offset

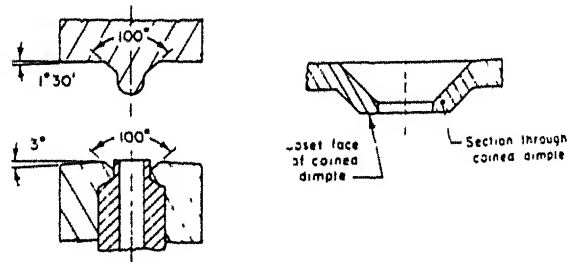


Deep drawing : Deep drawing is a process in which , sheet blank (hot or cold) is usually subjected to a peripheral hold down pressure, and the blank is forced by a punch into and through a die to form a deep recessed part having a wall thickness almost the same as that of the work



Deep drawing: (A) first draw, (B) redraw, (C) reverse draw

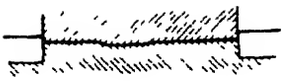
Dimpling : Dimpling is a process for producing small conical flanges around holes in sheet metal parts that are to be assembled with flush or flat headed rivets. This is most commonly applied to sheets that are too thin for counter sinking



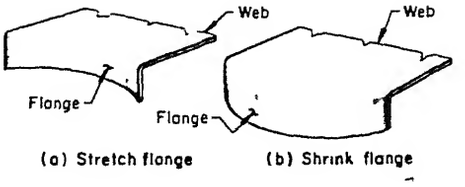
(a) Tooling setup

(b) Dimpled part

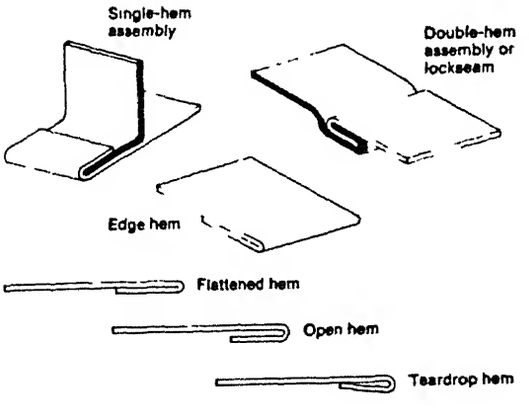
Embossing : Embossing is a process of producing raised or sunken designs in sheet material by means of male and female dies, theoretically with no metal thickness.



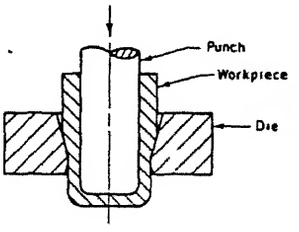
Flanging : Flanging process is similar to bending, except that the metal bent down is shorter, when compared to the overall part size Various types of flanging are straight, contoured, stretch, shrink, and reverse.



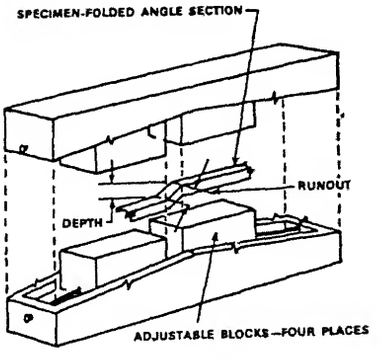
Hemming : Hemming is an operation in which flanges are flattened against the workpiece in 180 degree bends to make a finished or reinforced edge. Different types of hems are closed, open, and teardrop



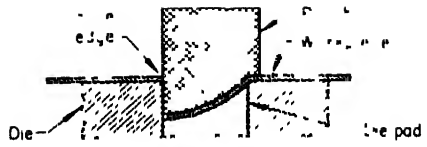
Ironing : Ironing is a process of smoothing and thinning the wall of a shell or cup (cold or hot), while retaining the original thickness of the bottom by forcing the shell through a die with a punch.



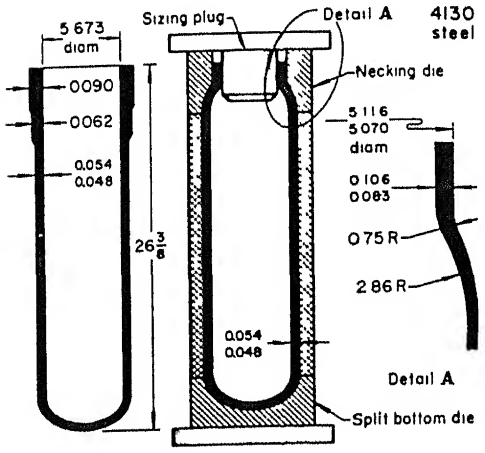
Joggling : In this process, an offset is obtained in a flat plane by two parallel bends in opposite directions at the same angle.



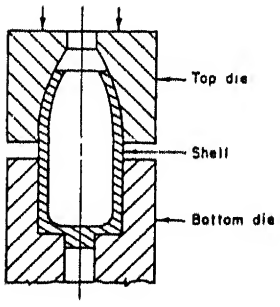
Louvering : Louvering is a process in which both the operations lancing and bending of the part are done



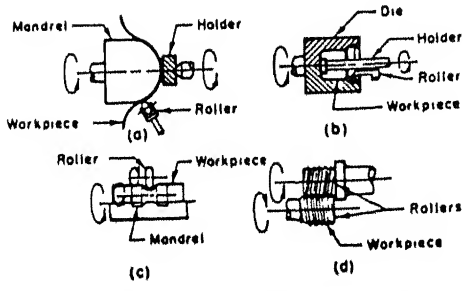
Necking : Necking is a process in which the diameter of a drawn cup or shell is reduced. The work metal is forced into compression by die reduction method, which results in an increase in both length and wall thickness of a part



Nosing : Nosing is a (cold or hot) forming process of tapering or rounding the end of a shell or tubular component by axial pressing with a shaped die.

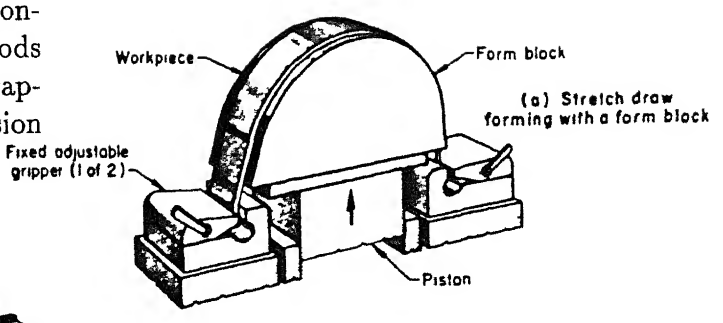


Spinning : Spinning is a process of shaping seamless dished parts by the combined forces of rotation and pressure, without changing the thickness of raw material. Various types of spinning operations are hollow shaping, bulging, reducing, and threading.

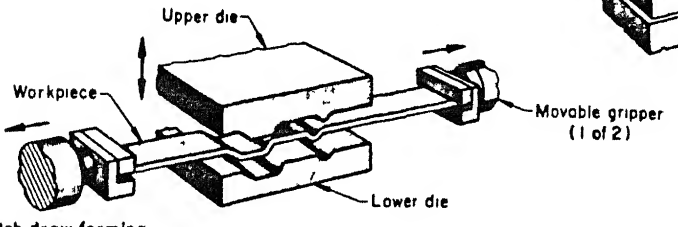


(a) hollow shape, (b) bulging, (c) reducing, (d) threading

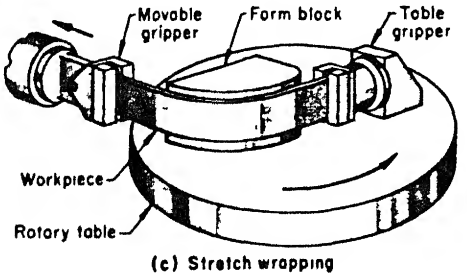
Stretch forming : Stretch forming is a process for obtaining the required shape from a blank by holding the workpiece under tension The work material is stretched beyond its yield point, so that the contour is retained There are four methods viz., stretch draw forming, stretch wrapping(rotary stretch forming), compression forming, and radial draw forming



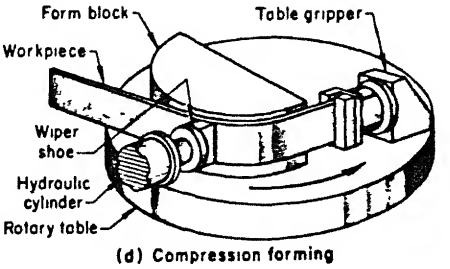
(a) Stretch draw forming with a form block



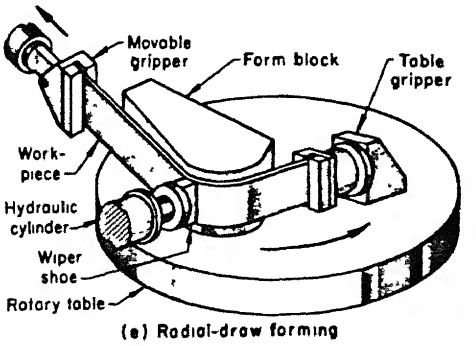
(b) Stretch draw forming with mating dies



(c) Stretch wrapping

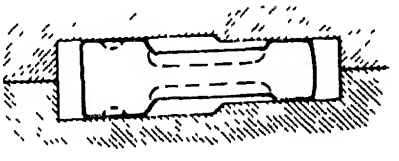


(d) Compression forming



(e) Radial-draw forming

Sizing : The metal is entirely free to flow, but the die comes together on solid contact faces in order to make the accuracy obtained less dependent on the thickness and hardness of the original forging



Appendix C

Definitions of Set theory

Basic concepts

This appendix gives the definitions of set theory related to the present work for easy reference (Lipshutz, 1981, Singer, 1969; Zied 1991) In this thesis, a set is represented in both roster and descriptive methods. In roster method, a set F is said to be *equal to the set of elements* f_1, f_2, f_3, f_4, and is represented as

$$F = \{f_1, f_2, f_3, f_4\}.$$

In case of descriptive method, set F is read as F *is equal to the the set of elements* f_j, *such that* f_j *equals* f_1, f_2, f_3, f_4, and is represented as

$$F = \{f_j \mid f_j = f_1,\ f_2,\ f_3,\ f_4\}.$$

If a set has no element in it or is void, then the set is called an empty or null set and is represented by the symbol $\emptyset$.

Regardless of the set designation, set membership and non-membership is customarily indicated by $\in$ and $\notin$, respectively, i e , $f_1 \in F$, and $f_6 \notin F$ represents that element f_1 is a member (or element) of F, and f_6 is not a member of F.

Basic set operations

In set theory, the basic operations are union, intersection, difference and complement. The symbols used and their Venn diagrams are given in Fig. C.1.

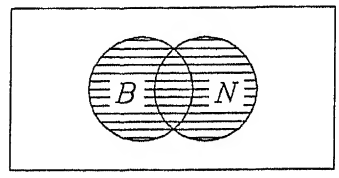
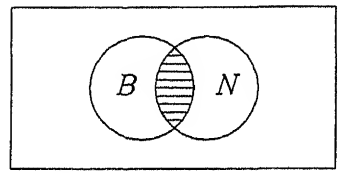
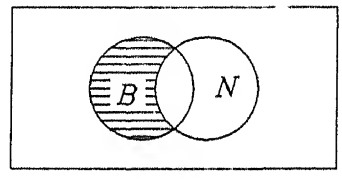
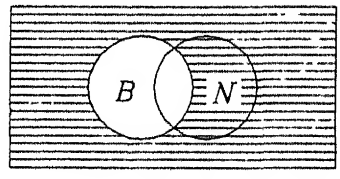
Name	Symbol	Venn diagram	Representation
Union	$\cup$		$B \cup N$
Intersection	$\cap$		$B \cap N$
Difference	$-$		$B - N$
Complement	$'$		B'

Figure C.1. Basic set operations

Partition sets

Let $\{F_j\}_{j\in J}$ be a family of non-empty subsets of F Then $\{F_j\}_{j\in J}$ is called *partition* of F if

1. $\bigcup_{j\in J} F_j = F$

2 For any $F_j, F_{j'}$,

$$\text{either } F_j = F_{j'} \text{ or } F_j \cap F_{j'} = \emptyset.$$

Example : Let $F = f_1, f_2, f_3, f_4, f_5$, and $R = f_1$, $B = f_2, f_4$, and $N = f_3, f_5$. Then R, B, N are said to be partition of F.

Randomly ordered set

A set is said to be randomly ordered, if the elements of it are not comparable.
Example : Let $V = v_1, v_2$. If the two elements v_1 and v_2 do not have any preceding relationships in a E^3 Euclidean space, then V is randomly ordered set.

Lexicographical ordering of set

A set is said to be lexicographically ordered, if the elements are arranged in the similar way as the alphabets are arranged in the dictionary.
Example : Let V be the vertex set in E^3 Euclidean space, having elements of ordered triplets, i.e.,

$$V = v_1, v_2, v_3, \ldots, v_n,$$

where $v_1 = x_1, y_1, z_1$; $v_2 = x_2, y_2, z_2$; .. , $v_n = x_n, y_n, z_n$. Therefore,

$$v_i = \{x_i, y_i, z_i \mid 1 \leq i \leq n\},$$

and

$$V = \{v_i \mid 1 \leq i \leq n\}.$$

The set V is said to be totally ordered, if

$$\begin{aligned} v_i \prec v_{i+1} \quad & \text{if } z_i < z_{i+1} \quad \text{or} \\ & \text{if } z_i = z_{i+1}, \quad \text{but } y_i < y_{i+1} \text{or} \\ & \text{if } z_i = z_{i+1} \quad \text{and } y_i = y_{i+1}, \text{ but } x_i < x_{i+1} \\ & \qquad\qquad \forall \quad 1 \leq i \leq n \end{aligned}$$

Then the set V is said to be lexicographically ordered.

Appendix D

Ray containment test

In the ray containment test, the intersection of line segments is used to determine whether an arbitrary point P (Fig. 4 2) lies inside or outside a given polygon (Burger and Gillies, 1990). The polygon is defined by a number of line segments or arc segments extending from vertex v_i to v_{i+1} with $i = 1, \ldots, p$ The polygon has p sides and, for convenience, the vertex v_{p+1} has been defined to be the same as v_1. For any given point $P(x, y)$, the containment of this point by the polygon can be determined by counting the intersections between the polygon edges and a horizontal line segment, $\vec{HL}$ The line $\vec{HL}$ is drawn in one direction to infinity and the number of times the line intersects the polygon edges are counted Mathematically, intersections of lines are counted as given below .

$$\vec{HL} = \vec{P} + \mu \vec{i} \qquad (\mu > 0) \tag{D.1}$$

where $\vec{i}$ is a unit vector in the X direction and μ is a scalar constant indicating the location of the line $\vec{HL}$

Let L_1 be a line segment of a polygon given by

$$\vec{L}_1 = \vec{V}_i + \nu (\vec{V}_{i+1} - \vec{V}_i) \qquad \forall\ i = 1, \ldots, p \tag{D.2}$$

where ν is a scalar constant indicating the section of a line segment of the polygon considered.

The intersection of the horizontal ray and line segment is expressed by the vector equation

$$\vec{HL} = \vec{L}_1.$$

Thus,

$$\vec{P} + \mu \vec{i} = \vec{V} + \nu (\vec{V}_{i+1} - \vec{V}_i) \qquad \forall\ i = 1, \ldots, p \tag{D 3}$$

The above relationship can be written for each of the two cartesian coordinates as follows :

$$x + \mu = x + \nu (x_{i+1} - x_i) \tag{D 4(a)}$$

$$y = y_i + \nu\,(y_{i+1} - y_i) \qquad \forall\ i = 1,\ \ldots,\ p. \qquad (D\ 4(b))$$

The above equation is solved for μ and ν. If $\mu > 0$ and $0 \leq \nu \leq 1$, then horizontal line $\vec{HL}$ has intersected the edge (made of v_i and v_{i+1}) of the polygon under consideration. The number of times the line crosses the polygon edges are counted. If the count is *odd*, then the point $P(x, y)$ is contained and if it is *even*, then the point $P(x, y)$ is outside the polygon.

References

AGARWAL, S.C., and WAGGENSPACK Jr, W.N., 1992, Decomposition method for extracting face topologies from wire frame models., *Computer Aided Design*, **24**, 3, 123-140.

ALTAN, T., OH, S.I., and GEGEL, H., 1983, Metal forming : fundamentals and applications, ASM International, Metals Parks, OH.

ARMSTRONG, G.T., CAREY, G.C., and DE PENNINGTON, A , 1984, Numerical code generation from a geometric modeling system, *Solid Modeling by Computers from Theory to Applications*, (PICKETT, M.S., and BOYSE, J.W., eds.), Plenum Press, 139-157.

AutoCAD Release-10, Reference manual, 1989, AutoDesk Inc

BURGER, P., and GILLIES, D., 1990, Interactive computer graphics: functional, procedural and Device level methods, Addison-Wesley publishing company, Inc., Great Britain.

CHANG, T.C., 1990, Expert process planning for manufacturing, Addison-Wesley publishing company, Inc., USA.

CHAUNG, S.H. and HENDERSON, M.R., 1990, Three dimensional shape pattern recognition using vertex classification and vertex edge graphs, *Computer Aided Design*, **22**, 6, 377-387.

CORNEY, J., and CLARK, D.E.R., 1991, Method for finding holes and pockets that connect multiple faces in 2 1/2 D objects, *Computer Aided Design*, **23**, 10, 658-668.

de VIN, L J., VRIES DE, J., STREPPEL, A H , KLAASSEN, E J W., and KALS, H J.J., 1994, The generation of bending sequences in a CAPP system for sheet metal components, *Journal of Materials Processing Technology*, **41**, 331 - 339.

DEO, N., 1990, Graph theory with applications to engineering and computer science, Prentice–Hall of India Private Limited, New Delhi.

EHRISMANN, R , and REISSNER, J., 1988, Intelligent manufacture of laser cutting, punching and bending parts *Robotics and Computer Integrated Manufacturing*, **4**, 3/4, 511–515.

FERREIRA, J C E., and HINDUJA, S., 1990, Convex hull based feature recognition method for 2.5 D components, *Computer Aided Design*, **22**, 1, 41–49.

GALLAGHER, C.C., and KNIGHT, W.A., 1973, Group technology, Butterworth, London.

GANTER, M. A , and SKOGLUND, P. A., 1993, Feature extraction for casting and core development, *Transactions of ASME*, **115**, December, 744–750.

GAVANKAR, P., 1993, Graph based recognition of morphological features, *Journal of Intelligent Manufacturing*, **4**, 209–218.

GAVANKAR, P., and HENDERSON, M.R., 1990, Graph- based extraction of protrusions and depressions from boundary representations, *Computer Aided Design*, **22**, 7, 442–450.

GRAVES, G.R., SAHAY, A., and PARKS, C.M., 1989, The role of feature recognition in production planning and a proposed approach, *Computer Integrated Manufacturing Systems*, **2**, 3.

GROOVER, M P., 1992, Automation, production systems, and computer integrated manufacturing, Prentice–Hall of India Private Limited, New Delhi.

HENDERSON, M. R., and ANDERSON, D., 1984, Computer recognition and extraction of form features : a CAD/CAM link, *Computer in Industry*, **5**, 329–339.

HOFFMANN, M., GEISSLER, U., and GEIGER, M., 1992, Computer aided generation of bending sequences for die-bending machines. *Journal of Materials Processing Technology*, **30**, 1–12.

IWATA, K., KAKINO, Y., OHBA, F , and SUGIMURA, N., 1980, Development of non-part family type computer aided production planning system CIMS/PRO, *Advanced Manufacturing Technology*, (BLAKE, P., ed.), 171–184, North-Holland.

JAKUBOWSKI, R , 1982, Syntactic characterization of machined part shapes, *Cybernetics and Systems*, **13**, 1–24

JOSHI, S , and CHANG, T.C., 1988, Graph based heuristic for recognition of machined features from a 3D solid model, *Computer Aided Design*, **20**, 2, 58–66.

JOSHI, S., VISSA, N N., and CHANG, T.C., 1988, Expert process planning system with solid interface, *International Journal of Production Research*, **26**, 5, 863–885.

KAKINO, Y., OHBA, F , MORIWAKI, T., and IWATA, K , 1977, A new method of parts description for computer aided production planning, *Advances in Computer-Aided Manufacturing*, (McPHERSON. D , ed.), 197–213, North-Holland

KANG, T, S , and NNAJI, B.O., 1993, Feature representation and classification for automatic process planning system *Journal of Manufacturing Systems*, **12**, 2, 133–145.

KANG, T., and WOO, T., 1991, Algorithmic aspects of alternating sum of volumes part 1: data structure and difference operation, *Computer Aided Design*, **23**, 5, 357–366

KANG, T., and WOO, T., 1991, Algorithmic aspects of alternating sum of volumes part 2: non convergence and its remedy, *Computer Aided Design*, **23**, 6, 435 –443.

LASCOE, O. D., 1988, Handbook of fabrication process, ASM International, Metals Parks, Ohio, USA.

LENAU, T., and LEO, A., 1990, Prerequisites for CAPP, *presented at the CIRP seminar on manufacturing systems*, Univ. of twenth, Netherlands, 1–8.

LENAU, T., and MU, L., 1991, Features in integrated modeling of products and their production, Discussion paper for the symposium/workshop on feature based approaches to design and process planning, Loughborough, UK.

LENTZ, D. H., and SOWERBY, R., 1993, Feature extraction of concave and convex regions and their intersections, *Computer Aided Design*, **25**, 7, 421–437.

LENTZ, D. H., and SOWERBY, R., 1994, Hole extraction of sheet metal components, *Computer Aided Design*, **26**, 10, 771–783.

LEOU, J. J., and TSAI, W. H., 1987, Automatic rotational symmetry determination for shape analysis, *Pattern Recognition*, **20**, 6, 571–582

LI, R , 1988, A part feature recognition system for rotational parts, *International Journal of Production Research*, **26**, 9, 1451–1475

LIN, Z C., and PEING, G J , 1994, An investigation of an expert system for sheet metal bending design, *Journal of Materials Processing Technology*, **43**, 165–176.

LIPSCHUTZ, S , 1981, Set theory and related topics, Schaum's outline series, McGraw–Hill book company, Singapore.

LOH, H.T., NEE, A.Y.C., and CHONG, T.C., 1992, Computer generated flat pattern development of heating, ventilating and air ductings, *Journal of Materials Processing Technology*, **29**, 173–189.

MAREFAT, M., FEGHHI, S.J and KASHYAP, R.L., 1990, IDP: Automating the CAD/CAM link by reasoning about shape, *Proceedings of 6th IEEE conference on Artificial Intelligence*, Santa barbara California, March, 1–8.

MAREFAT, M., and KASHYAP, R.L., 1990, Geometric reasoning for recognition of three dimensional object features, *IEEE transactions on pattern analysis and machine intelligence*, **12**, 10, 949–965.

MEERAN, S., and PRATT, M. J., 1993, Automated feature recognition from 2–D drawings, *Computer Aided Design*, **25**, 1, 7–17.

MEINEL, M P., 1945, Formed sheet-metal parts classified by shape, *Production Engineering*, **16**, 167, 241, 334.

METALS HANDBOOK, 1979, *Forming*, eighth edition, **vol.4.**,American Society for metals, Metals Parks, OH.

MILACIC, V. R., 1985, SAPT–Expert system for manufacturing process planning, *Computer Aided Intelligent Process Planning*, (LIU, C.R., CHANG, T.C., and KOMANDURI, R.), **19**, ASME.

NEE, A.Y.C., and CHONG, T.C., 1988, A micro computer based flat development, *Sheet Metal Industry*, **65**, 92–93.

NNAJI, B.O., KANG, T. S., YEH, S., and CHEN, J. P., 1991, Feature reasoning for sheet metal components *International Journal of Production Research*, **29**, 9, 1867–1896.

OKINO, N , and KUBO, H., 1970, Technical information system for computer aided design, drawing, and manufacturing, Proceedings of the second PROLAMAT.

PALANI, R , WAGONER, R H., and NARASIMHAN, K , 1994, A knowledge based simulations approach for sheet metal forming, *Journal of Materials Processing Technology*, **45**, 703 – 708

PICKUP, F , and PARKER, M.A , 1972, Engineering drawing with worked examples, Version 2, Hutchinson Educational Limited, London.

PRASAD, Y K. D. V., and SOMASUNDARAM, S., 1993, A mathematical model for bend allowance calculation in automated sheet metal bending, *Journal of Materials Processing Technology*, **39**, 3–4, 337 –356

RAGGENBASS, A., and REISSNER, J., 1989, Stamping– laser combination in sheet processing *Annals of CIRP*, **38**, 1, 291–294

RAGGENBASS, A., and REISSNER, J., 1991, Automatic generation of NC production plans in stamping and laser cutting *Annals of CIRP*, **40**, 1, 247–250

ROGERS, D. F., and ADAMS, J. A., 1990, Mathematical elements for computer graphics, McGraw–Hill Book Co , Singapore.

REHG J.A., 1994, Computer integrated manufacturing, Prentice Hall Career & Technology, Englewood Cliffs, NJ, USA.

SACHS, G , and VOEGELI, H.E., 1966, Principles and methods of sheet–metal fabricating, Reinhold publishing corporation, New York, USA.

SAHAY, A., GRAVES, G.R., PARKS, C.M., and MANN, L., 1990, A methodology for recognizing features in two dimensional cylindrical part designs. *International Journal of Production Research*, **28**, 8, 1401–1416.

SATYADEV, A., 1995, Flat pattern development of bending and deep drawing of sheet metal components, M.Tech, thesis (unpublished), IIT–Kanpur, India.

SHAH, J. J., 1991, Assessment of features technology, *Computer Aided Design*, **23**, 5, 331–343.

SHAH, J J , and ROGERS, M T , 1988, Expert form feature modeling shell, it Computer Aided Design, **20**, 9, 515 – 524.

SHPITALNI, M., and FISCHER, A., 1991, CSG representation as a basis for extraction of machining features, *Annals of CIRP*, **40**, 1, 157–160.

SCHULTE, R.M , PADMANABHAN, S., and DEVGUN, M S , 1992, Feature driven process based approach to the integration of CAD/CAM in wire frame models, *International Journal of Production Research*, **30**, 5, 1005 –1028.

SINGER, I.M , 1969, Lecture notes on elementary topology and geometry, Scott Foresman and Company, Glenview, Illonois, USA.

SMITH, S.J., COHEN, H.P., DAVIS, W.J , and IRANI, A.S., 1992, Process plan for sheet metal parts using an integrated feature based expert system approach *International Journal of Production Research*, **30**, 5, 1175–1190.

SOLOMONS, O. W., VAN HOUTEN, F. J. A M., and KALS H. J. J., 1993, Review of research in feature based design, *Journal of Manufacturing Systems*, **12**, 2, 113 –132.

SRINIVASAN, R., and LIU, C.R., 1985, Extraction of manufacturing details from geometric models, *Computers and Industrial Engineering*, **9**, 125–133.

TILLEY, S., 1992, Integration of CAD/CAM and production control for sheet components manufacturing, *Annals of CIRP* , **41**, 1, 177–180.

van't ERVE, A.H., and KALS, H.J.J., 1986, XPLANE – a generative computer aided process planning system for part manufacturing, *Annals of CIRP*, bf 35, 1, 325–329.

VIDLICKA, I.P., 1993, Computer aided manufacture of sheet metal components, *Sheet Metal Industries*, June, 22–25.

WANG, H., and WYSK, R.A., 1988, AIMSI: a prelude to a new generation of integrated CAD/CAM systems, *International Journal of Production Research*, **26**, 1, 119–131.

WEXLER, C., 1962, Analytic Geometry : A vector approach, Addison–Wesley Publishing company, Inc., USA.

WILSON, P.R., 1985, Euler formula and geometric modeling, *IEEE Computer Graphics and Application*, August, 24–36.

WOODWORK, J.R., 1988, Some speculations on feature recognition, *Computer Aided design*, **20**, 4, 189–196

WOO, T C., 1982, Feature extraction by volume decomposition, Conference on CAD/CAM Technology in Mechanical Engineering, M I T., March.

ZEID, I , 1991, CAD/CAM theory and practice, McGraw–Hill, Inc., Singapore.

ZOPF, R., 1988, CAD for sheet metal parts, *Sheet Metal Industry*, **65**, 11, 584.

Bibliography

CHANG, T.C , and WYSK, R A., 1985, An introduction to automated process planning systems, Prentice–Hall, Englewood Cliffs, New Jersey, USA.

ALTING, L., and ZHANG, H., 1989, Computer aided process planning : the stat-of-the-art survey, *International Journal of Production Research*, **27**, 4, 553–585.

MANTYLA, M., 1988, An introduction to solid modeling, Computer Science Press, Inc., Maryland, USA.

MORTENSEN, M.E., 1985, Geometric modeling, John Wiley.

MUFTI, A A., MORRIS, M. L., and SPENCER, W B., 1990, Data exchange standards for computer aided engineering and manufacturing, *International Journal of Computer Applications in Technology*, **3**, 2, 70–80.

NIEVERGELT, J., and HINRICHS, K. H., 1993, Algorithms and data structures : with applications to graphics and geometry, Prentice–Hall, Englewood Cliffs, New Jersey, USA.

PREPARATA, F P. and SHAMOS, M I., 1985, Computational geometry : an introduction, Springer–Verlag, New York, USA.

STATELY, S.M., HENDERSON, M.R., and ANDERSON, D.C., 1983, Using syntactic pattern recognition to extract information from solid geometric data base, *Computers in Mechanical engineering*, **2**, 61 –66

TENENBAUM, A. M., LANGSAM, Y., and AUGENSTEIN, M. J., 1992, Data structures using C, Prentice–Hall of India Private Limited, New Delhi, India.

WENTLAND, K , and DUTTA, D., 1993, Method for offset–curve generation for sheet metal design, *Computer Aided Design*, **25**, 10, 662–670.